Marzena Kryszkiewicz
Henryk Rybinski
Andrzej Skowron
Zbigniew W. Raś (Eds.)

LNAI 6804

Foundations of Intelligent Systems

19th International Symposium, ISMIS 2011
Warsaw, Poland, June 2011
Proceedings

Springer

Lecture Notes in Artificial Intelligence 6804

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

Subseries of Lecture Notes in Computer Science

Marzena Kryszkiewicz Henryk Rybinski
Andrzej Skowron Zbigniew W. Raś (Eds.)

Foundations of Intelligent Systems

19th International Symposium, ISMIS 2011
Warsaw, Poland, June 28-30, 2011
Proceedings

Springer

Volume Editors

Marzena Kryszkiewicz
Henryk Rybinski
Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
E-mail: {mkr, hrb}@ii.pw.edu.pl

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

Zbigniew W. Raś
University of North Carolina, Charlotte, NC 28223, USA
and Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
E-mail: ras@uncc.edu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21915-3 e-ISBN 978-3-642-21916-0
DOI 10.1007/978-3-642-21916-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011929786

CR Subject Classification (1998): I.2, H.3, H.2.8, H.4-5, I.4, F.1, I.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the papers selected for presentation at the 19th International Symposium on Methodologies for Intelligent Systems—ISMIS 2011, held in Warsaw, Poland, June 28-30, 2011. The symposium was organized by the Institute of Computer Science at Warsaw University of Technology. ISMIS is a conference series that started in 1986. Held twice every three years, ISMIS provides an international forum for exchanging scientific, research, and technological achievements in building intelligent systems.

The following major areas were selected for ISMIS 2011: theory and applications of rough sets and fuzzy sets, knowledge discovery and data mining, social networks, multi-agent systems, machine learning, mining in databases and warehouses, text mining, theoretical issues and applications of intelligent Web, applications of intelligent systems, inter alia in sound processing, biology and medicine.

Out of 131 submissions, 71 contributed papers were accepted for publication by the international Program Committee with help of additional external referees. Every paper was assigned to three reviewers. Initially, some of these papers were conditionally approved subject to revision and then re-evaluated. In addition, four plenary talks were given by Jaime Carbonell, Andrzej Czyżewski, Donato Malerba, and Luc De Raedt. Four special sessions were organized: Special Session on Rough Sets, devoted to the Memory of Zdzisław Pawlak, Special Session on Challenges in Knowledge Discovery and Data Mining, devoted to the Memory of Jan Żytkow, Special Session on Social Networks, and Special Session on Multi-Agent Systems.

The ISMIS conference was accompanied by the data mining contest on Music Information Retrieval, and Industrial Session on Emerging Intelligent Technologies in Industry, as well as a post-conference workshop, devoted to SYNAT, which is a large scientific Polish project funded by the National Centre for Research and Development (NCBiR), aiming at creating a universal hosting and scientific content storage and sharing platform for academia, education, and open knowledge society.

We wish to express our thanks to all the ISMIS 2011 reviewers, and to the invited speakers. Our thanks go to the organizers of special sessions, namely, Jerzy Grzymała-Busse (Special Session on Rough Sets), Shusaku Tsumoto (Special Session on Challenges in Knowledge Discovery and Data Mining), Hakim Hacid (Special Session on Social Networks), and Barbara Dunin-Kęplicz (Special Session on Multi-Agent Systems).

We would also like to express our appreciation to the organizers of accompanying events: Marcin Wojnarski and Joanna Świetlicka from TunedIt, who successfully launched the contest; Bożena Kostek, Paweł Żwan, Andrzej Sitek, and Andrzej Czyżewski for providing a data set for the Music Genres contest

task; Zbigniew Raś and Wenxin Jiang for providing a data set for the Music Instruments task; Dominik Ryżko for his involvement in all the organizational matters related to ISMIS 2011, and for organizing the industrial session; Robert Bembenik and Łukasz Skonieczny for organizing the post-conference SYNAT workshop. We are grateful to Piotr Kołaczkowski for the creation and maintenance of the conference website, as well as Bożenna Skalska and Joanna Konczak for their administrative work.

Our sincere thanks go to Aijun An, Petr Berka, Jaime Carbonell, Nick Cercone, Tapio Elomaa, Floriana Esposito, Donato Malerba, Stan Matwin, Jan Rauch, Lorenza Saitta, Giovanni Semeraro, Dominik Ślęzak, Maria Zemankova, who served as members of ISMIS 2011 Steering Committee. Moreover, our thanks are due to Alfred Hofmann of Springer for his continuous support and to Anna Kramer and Ingrid Haas for their work on the proceedings.

June 2011

Marzena Kryszkiewicz
Henryk Rybiński
Andrzej Skowron
Zbigniew W. Raś

Organization

ISMIS 2011 was organized by the Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland.

Executive Committee

General Chair	Zbigniew Raś (University of North Carolina at Charlotte, USA and Warsaw University of Technology, Poland)
Conference Chair	Marzena Kryszkiewicz (Warsaw University of Technology, Poland)
Program Co-chairs	Henryk Rybiński (Warsaw University of Technology, Poland)
	Andrzej Skowron (University of Warsaw, Poland)
Organizing Chair	Dominik Ryżko (Warsaw University of Technology, Poland)

Steering Committee

Aijun An	York University, Canada
Petr Berka	University of Economics, Prague, Czech Republic
Jaime Carbonell	Carnegie Mellon University, USA
Nick Cercone	York University, Canada
Tapio Elomaa	Tampere University of Technology, Finland
Floriana Esposito	University of Bari, Italy
Donato Malerba	University of Bari, Italy
Stan Matwin	University of Ottawa, Canada
Zbigniew Raś	University of North Carolina at Charlotte, USA and Warsaw University of Technology, Poland
Jan Rauch	University of Economics, Prague, Czech Republic
Lorenza Saitta	Università del Piemonte Orientale, Italy
Giovanni Semeraro	University of Bari, Italy
Dominik Ślęzak	Infobright Inc., Canada and University of Warsaw, Poland
Maria Zemankova	NSF, USA

Program Committee

Luigia Carlucci Aiello	Università di Roma La Sapienza, Italy
Troels Andreasen	Roskilde University, Denmark
Jan Bazan	University of Rzeszów, Poland
Salima Benbernou	Université Paris Descartes, France
Marenglen Biba	University of New York, Tirana, Albania
Maria Bielikova	Slovak University of Technology in Bratislava, Solvakia
Ivan Bratko	University of Ljubljana, Slovenia
Francois Bry	University of Munich, Germany
Cory Butz	University of Regina, Canada
Jacques Calmet	Universität Karlsruhe, Germany
Longbing Cao	University of Technology Sydney, Australia
Sandra Carberry	University of Delaware, USA
Michelangelo Ceci	Università di Bari, Italy
Jianhua Chen	Louisiana State University, USA
Bruno Cremilleux	University of Caen, France
Juan Carlos Cubero	University of Granada, Spain
Alfredo Cuzzocrea	University of Calabria, Italy
Andrzej Czyżewski	Gdańsk University of Technology, Poland
Agnieszka Dardzinska	Bialystok University of Technology, Poland
Nicola Di Mauro	Università di Bari, Italy
Włodzisław Duch	Nicolaus Copernicus University
Barbara Dunin-Kęplicz	University of Warsaw, Poland and Polish Academy of Sciences, Poland
Peter W. Eklund	The University of Wollongong, Australia
Edward Fox	Virginia Polytechnic Institute and State University, USA
Piotr Gawrysiak	Warsaw University of Technology, Poland
Attilio Giordana	Università del Piemonte Orientale, Italy
Salvatore Greco	University of Catania, Italy
Jerzy Grzymała-Busse	University of Kansas, USA
Fabrice Guillet	Université de Nantes, France
Mohand-Said Hacid	Université Claude Bernard Lyon 1, France
Hakim Hacid	Bell Labs, Alcatel-Lucent, France
Allel Hadjali	IRISA/ENSSAT, Université de Rennes 1, France
Mirsad Hadzikadic	University of North Carolina at Charlotte, USA
Howard Hamilton	University of Regina, Canada
Perfecto Herrera	Universitat Pompeu Fabra, Spain
Shoji Hirano	Shimane University, Japan
Władysław Homenda	Warsaw University of Technology, Poland
Jimmy Huang	York University, Canada

Nathalie Japkowicz	University of Ottawa, Canada
Janusz Kacprzyk	Polish Academy of Sciences, Poland
Józef Kelemen	Silesian University, Czech Republic
Mieczysław Kłopotek	Polish Academy of Sciences, Poland
Jacek Koronacki	Polish Academy of Sciences, Poland
Bożena Kostek	Gdańsk University of Technology, Poland
Stanisław Kozielski	Silesian University of Technology, Poland
Marzena Kryszkiewicz	Warsaw University of Technology, Poland
Lotfi Lakhal	Aix-Marseille Université, France
Patrick Lambrix	Linköping University, Sweden
Rory Lewis	University of Colorado at Colorado Springs, USA
Chao-Lin Liu	National Chengchi University, Taiwan
Jiming Liu	Hong Kong Baptist University PR China
Pasquale Lops	University of Bari, Italy
Michael Lowry	NASA, USA
Ramon López de Mántaras	CSIC in Barcelona, Spain
David Maluf	NASA, USA
Tomasz Martyn	Warsaw University of Technology, Poland
Sally McClean	Ulster University, Ireland
Paola Mello	University of Bologna, Italy
Mikołaj Morzy	Poznan University of Technology, Poland
Tadeusz Morzy	Poznan University of Technology, Poland
Hiroshi Motoda	Osaka University and AFOSR/AOARD, Japan
Mieczysław Muraszkiewicz	Warsaw University of Technology, Poland
Neil Murray	University at Albany - SUNY, USA
Pavol Navrat	Slovak University of Technology in Bratislava, Slovakia
Hung Son Nguyen	University of Warsaw, Poland
Sinh Hoa Nguyen	Polish-Japanese Institute of Information Technology, Poland
Nicola Orio	University of Padova, Italy
Petra Perner	IBAI Leipzig, Germany
James Peters	University of Manitoba, Canada
Jean-Marc Petit	University of Lyon, France
Olivier Pivert	University of Rennes 1, France
Jaroslav Pokorny	Charles University, Czech Republic
Lech Polkowski	Polish-Japanese Institute of Information Technology, Poland
Henri Prade	University Paul Sabatier, France
Grzegorz Protaziuk	Warsaw University of Technology, Poland
Seppo Puuronen	University of Jyväskylä, Finland
Vijay Raghavan	University of Louisiana at Lafayette, USA
Naren Ramakrishnan	Virginia Tech, USA

Sheela Ramana	University of Winnipeg, Canada
Zbigniew Raś	University of North Carolina at Charlotte, USA
Gilbert Ritschard	University of Geneva, Switzerland
Henryk Rybinski	Warsaw University of Technology, Poland
Dominik Ryżko	Warsaw University of Technology, Poland
Hiroshi Sakai	Kyushu Institute of Technology, Japan
Andrzej Skowron	The University of Warsaw, Poland
Roman Słowiński	Poznan University of Technology, Poland
Vaclav Snasel	VSB-Technical University of Ostrava, Czech Republic
Nicolas Spyratos	University of Paris South, France
Jerzy Stefanowski	Poznan Univeristy of Technology, Poland
Jarosław Stepaniuk	Białystok University of Technology, Poland
Olga Stepankova	Czech Technical Univeristy in Prague, Czech Republic
Zbigniew Suraj	University of Rzeszów, Poland
Vojtech Svatek	University of Economics, Prague, Czech Republic
Piotr Synak	Polish-Japanese Institute of Information Technology, Poland
Marcin Szczuka	University of Warsaw, Poland
Paweł Terlecki	Microsoft Corp., USA
Krishnaprasad Thirunarayan	Wright State University, USA
Li-Shiang Tsay	North Carolina A&T State University, USA
Shusaku Tsumoto	Shimane University, Japan
Athena Vakali	Aristotle University of Thessaloniki, Greece
Christel Vrain	LIFO - University of Orléans, France
Alicja Wakulicz-Deja	University of Silesia, Poland
Krzysztof Walczak	Warsaw University of Technology, Poland
Guoyin Wang	Chongqing University of Posts and Telecommunications, China
Rosina Weber	Drexel University, USA
Alicja Wieczorkowska	Polish-Japanese Institute of Information Technology, Poland
Marek Wojciechowski	Poznan University of Technology, Poland
Jakub Wróblewski	The University of Warsaw, Poland
Wensheng Wu	University of North Carolina at Charlotte, USA
Xindong Wu	University of Vermont,USA
Yiyu Yao	University of Regina, Canada
Sławomir Zadrożny	Polish Academy of Sciences, Poland
Yan Zhang	University of Western Sydney, Australia
Ning Zhong	Maebashi Institute of Technology, Japan
Djamel Zighed	ERIC, University of Lyon 2, France

Reviewers of Special Sessions

Armen Aghasaryan	Alcatel-Lucent Bell Labs, France
Petr Berka	University of Economics, Prague, Czech Republic
Francesco Bonchi	Yahoo! Research, Spain
Chien-Chung Chan	University of Akron, USA
Davide Ciucci	Università di Milano Bicocca, Italy
Samik Datta	Bell Labs, Bangalore, India
Ludovic Denoyer	LIP6 - University of Paris 6, France
Marcin Dziubiński	The University of Warsaw, Poland
Amal El Fallah Seghrouchni	LIP6 - University of Pierre and Marie Curie, France
Paul El Khoury	University of Lyon 1, France
Cécile Favre	University of Lyon 2, France
Maria Ganzha	University of Gdańsk, Poland
Yannet Interian	Google, USA
Masahiro Inuiguchi	Osaka University, Japan
Marek Kisiel-Dorohinicki	AGH University of Science and Technology, Poland
Jan Komorowski	University of Uppsala, Sweden
David Konopnicki	IBM Haifa Research Lab, Israel
Shonali Krishnaswamy	Monash University, Australia
Tsau Young Lin	San Jose State University, USA
Yosi Mass	IBM Haifa Research Lab, Israel
Helen Paik	University of New South Wales, Australia
Jan Rauch	University of Economics at Prague, Czech Republic
Wojciech Rząsa	Rzeszów University, Poland
Sherif Sakr	The University of New South Wales, Australia
Hyoseop Shin	Konkuk University, Korea
Julia Stoyanovich	University of Pennsylvania, USA
Aixin Sun	Nanyang Technological University, Singapore
Einoshin Suzuki	Kyushu University, Japan
Dirk Van Den Poel	Ghent University, Belgiam
Rineke Verbrugge	University of Groningen, The Netherlands
Takashi Washio	ISIR, Osaka University, Japan
Anita Wasilewska	Stony Brook University, USA
Urszula Wybraniec-Skardowska	Poznań School of Banking, Poland
Katsutoshi Yada	Data Mining Applied Research Center, Japan
Jing Tao Yao	University of Regina, Canada
Tetsuya Yoshida	Hokkaido Univerisity, Japan
Wojciech Ziarko	University of Regina, Canada

External Reviewers

Andrzejewski, Witold
Ayhan, Murat Seckin
Bembenik, Robert
Benouaret, Karim
Bordino, Ilaria
Bragaglia, Stefano
Casali, Alain
Chesani, Federico
De Gemmis, Marco
Dr. Wenxin Jiang
Ellwart, Damian
Gabadinho, Alexis
Goodall, Peter
Groznik, Vida
Gryz, Jarek
Gurram, Mohana
Henry, Christopher
Janež, Tadej
Knight, Christopher
Košmerlj, Aljaž
Kożuszek, Rajmund
Laurent, Dominique
Loglisci, Corrado
Marinica, Claudia
Matusiewicz, Andrew
Mozina, Martin
Muller, Nicolas S.
Musaraj, Kreshnik
Navigli, Roberto
Nedjar, Sébastien
Nonino, Fabio
Nowak, Robert
Ouziri, Mourad
Phan-Luong, Viet
Poncelet, Pascal
Putz, Peter
Rogovschi, Nicoleta
Roitman, Haggai
Rosati, Riccardo
Sitek, Andrzej
Skonieczny, Łukasz
Sottara, Davide
Studer, Matthias
Taranto, Claudio
Wang, Yan
Woźniak, Stanisław
Zeng, Yi
Ziembinski, Radoslaw

Table of Contents

Invited Papers

Rough Sets - in Memoriam Zdzisław Pawlak

Challenges in Knowledge Discovery and Data Mining - in Memoriam Jan Żytkow

Social Networks

Multi-Agent Systems

Theoretical Backgrounds of AI

Machine Learning

Data Mining

Mining in Databases and Warehouses

Text Mining

Theoretical Issues and Applications of Intelligent Web

Application of Intelligent Systems in Sound Processing

Intelligent Applications in Biology and Medicine

Fuzzy Sets Theory and Applications

Intelligent Systems, Tools and Applications

Contest on Music Information Retrieval

Intelligent Multimedia Solutions Supporting Special Education Needs

Andrzej Czyzewski and Bozena Kostek

Multimedia Systems Department, Gdansk University of Technology,
Narutowicza 11/12, 80-233 Gdansk, PL
ac@pg.gda.pl

Abstract. The role of computers in school education is briefly discussed. Multimodal interfaces development history is shortly reviewed. Examples of applications of multimodal interfaces for learners with special educational needs are presented, including interactive electronic whiteboard based on video image analysis, application for controlling computers with facial expression and speech stretching audio interface representing audio modality. Intelligent and adaptive algorithms application to the developed multimodal interfaces is discussed.

1 Introduction

As regards the usage of PC computers in college classrooms, new research shows that they can actually increase students' engagement, attentiveness, participation and learning. However computer employed in the classroom may entail the following adverse effects:

- isolate school students,
- distract them from the teacher,
- break emotional links between pupils,
- prevent socializing during the lesson,
- change team work habits unfavorably,
- worsen eyesight acuity,
- influence negatively body posture.

Current research at the Multimedia Systems Department is intended to prove the following thesis: "technology developments can lead us toward a more natural way of using computers in general, especially in classrooms".

In order to let computers to be used in a more natural and spontaneous way, they should fulfill the following demands:

- their presence should remain unnoticed (for as much time as possible),
- they should provide fully open platform (in contrast to some recent restricted ones),
- should be operated in natural ways, e.g. by gestures,
- should interact with human senses much better.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 1–15, 2011.

In order to satisfy above demands, some new ways of human-computer-interfacing should be pursued. In turn, in the technology layer their realization demands solving problems requiring a simulation of intelligent thought processes, heuristics and applications of knowledge. In particular children with so called "special educational needs", i.e. children with various types of disabilities (communication senses problems, limb paralysis, infantile paralysis, partial mental retardation and others) can potentially benefit much from the availability of intelligent solutions helping them to learn using computers. The needs of partially impaired children motivated us at the Gdansk University of Technology to develop a prototype series of multimodal interfaces some of which are presented in this paper.

2 Multimodal Interfaces

The following milestones can be identified in the history of man-computing machine communication:

- in Ancient China the first known interface was the gills of abacus;
- in the 60's keyboards of cards perforator machines and teletypes appeared;
- when in the 70's the first terminals appeared, the sudden need for typing occurred, as terminals accepted only such form of input data;
- the first graphical operating system was developed in the 80's. This interface introduced us to the mouse – essentially a simple pointing device;
- the next stage were currently very popular graphical interfaces;
- fast evolution of computing power in the 90's allowed development of a fair speech and text recognition systems.

Still it is natural human tendency to speak, gesticulate and sometimes use handwriting, when communication is needed, thus various solutions in this domain appear on the market since 90's until present, including tablets with touch sensitive screen and others. Nowadays natural forms of communication are the most desired and interfaces using those are known as **multimodal interfaces**.

The subject is not new, however, so that many notions related to multimodal interfaces were hitherto conceived, including the following ones:

- Man-Machine Interaction (MMI) – (during II Word War);
- Human-Computer Interaction (HCI) – (in the 70's);
- Human-Machine Communication (HMC);
- Perceptual User Interface (PUI);
- Natural Interactive Systems (CityplaceNIS).

The term multimodal consists of two components, namely: multiplicity and modality, where modality it is the way of transferring and receiving information. There are several kinds of information in the communication, e.g.:

- natural language,
- hands gestures and movements,

– body language,
– facial expressions,
– handwriting style.

The multimodal systems can be divided into unimodal systems – those using only one modality e.g. speech recognition or text recognition or multimodal systems – those using several modalities as an input signal, e.g. speech recognition with simultaneous gesture capture. Applications of multimodal systems are widespread most in education to provide help to pupils with special needs, including:

– children with attention disorders (e.g. ADHD syndrome) – multimodal interfaces give the great opportunity to improve their learning skills through stimulating different senses helping to focus attention,
– concentration training – biofeedback usage,
– educational games with multimodal interaction,
– others (some cases will be presented later on).

Currently Multimedia Systems Department is carrying out several researches dealing with multimodal interfaces in a direct co-operation with industrial partners. Human senses: sight, hearing, touch and smell are involved. Moreover, gesture recognition with video camera image analysis is employed in many applications. The recognition based on image processing is nowadays research focus – much effort is put on eliminating the need of usage of all wire connections, sensors, gloves or other additional tools.

A common feature of all developed system is that their engineering demand undeterministic problem solving in algorithmic and especially signal processing layers. Thus, the technology layer realization demand solving problems requiring applications of heuristics, soft computing or in general: knowledge-base systems.

3 Gesture-Controlled Interactive Whiteboard

Interactive electronic whiteboards may support effectively for students who need to see the material presented again, or are absent from school, for struggling learners, and for children with special educational needs. The disadvantage of typical electronic whiteboards is their price which is partly the result of the necessity of using electronic pens and large frames equipped with sensors. To improve whiteboard content controlling in cases the system uses a camera, vision-based gesture recognition can be applied. Some attempts in this domain were presented in papers of others e.g. [1–3]. Authors of the latter paper use a portable projector for content displaying and a webcam. The equipment can be mounted on a helmet worn by the user. Special colorful tips are used on fingers to provide gesture controlling.

The system developed at the Multimedia Systems Department by M. Lech and B. Kostek [4] provides the functionality of electronic whiteboards and its essential feature is lack of the necessity of using any special manipulators or

sensors. Moreover, the whiteboard content can be handled (e.g. zoomed in / out, rotated) by dynamic hand gestures. Data gloves (cyber gloves) or special tips on fingers are not needed.

The hardware part of the system is presented in Fig. 1. It is composed of a PC (dual core), a multimedia projector, a webcam and a screen for projected image. The webcam is attached to the multimedia projector in such a way that both lenses are directed at the projection screen.

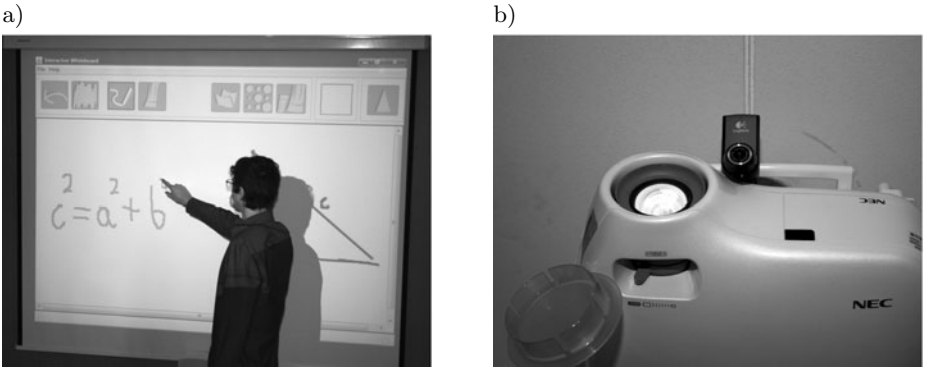


Fig. 1. Hardware part of the system: a) projection screen; b) multimedia projector coupled with a webcam [4]

3.1 Kalman Filtering

To provide reliable hand position tracking each captured frame is appropriately processed using Kalman filters. Considering the necessity of eliminating distortions introduced by camera lens, perspective transformations and impact of light on displayed image is performed. The image processing methods used have been described in earlier papers [4, 5].

Hand movements are modeled by motion vectors designated on a few successive camera frames. Each vector $\boldsymbol{u} = [u_x, u_y]$ is analyzed in the Cartesian coordinate system regarding velocity and direction (Fig. 2).

Two parameters of motion vectors, i.e. speed and direction, were used as a basis for gesture interpretation mechanism. Speed for motion vector within the time interval $t_i - t_{i-1}$, denoted as v_{ij}, where $j = i - 1$, was calculated according to Eq. (1). Direction for particular motion vector $\boldsymbol{u}_{ij} = [u_x^{ij}, u_y^{ij}]$ was denoted as an angle α_{ij} in relation to angle φ_{ij} between $\boldsymbol{u}_{ij}$ with origin at $[0, 0]$ and versor of y-axis, according to Eqs. (2) and (3).

$$v_{ij} = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \quad \left[\frac{px}{\mathrm{s} \cdot 10^{-1}}\right] \tag{1}$$

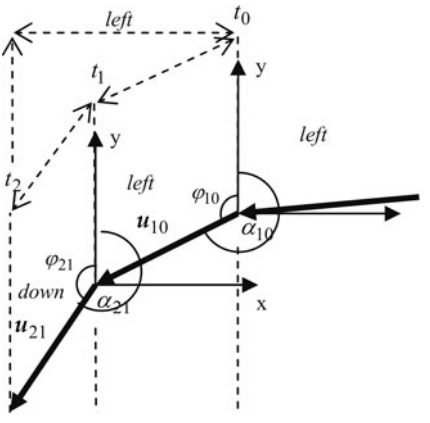


Fig. 2. Motion vectors created for semi-circular hand movement in the left direction

where: x_i and x_{i-1} are x positions of hand in time t_i and t_{i-1}, respectively, and y_i, y_{i-1} are y positions of hand in time t_i and t_{i-1}, respectively;

$$\varphi_{ij} = \frac{180° \cdot a_{ij} \cos \dfrac{u_y^{ij}}{|\boldsymbol{u}_{ij}|}}{\pi} \quad [°] \tag{2}$$

$$\alpha_{ij} = \begin{cases} \varphi_{ij}\,, & u_x^{ij} \geq 0 \\ 360° - \varphi_{ij}\,, & u_x^{ij} < 0 \end{cases} \tag{3}$$

For the velocity v_{ij}, also vertical and horizontal velocities are computed using trigonometric identities for angle φ_{ij} in relation to angle α_{ij}. The obtained horizontal and vertical velocities are expressed by Eqs. (4) and (5), respectively.

$$v_{ij}^x = v_{ij} \sin \alpha_{ij} \tag{4}$$

$$v_{ij}^y = v_{ij} \cos \alpha_{ij} \tag{5}$$

The Kalman filter [6] is used for estimating the state of a system from a series of noisy measurements. The predicted state $s_{t|t-1}$ in time t is related to state in time $t-1$ according to the following equation:

$$\hat{s}_{t|t-1} = F_t \hat{s}_{t-1|t-1} + w_{t-1} + B_{t-1} u_{t-1} \tag{6}$$

where F_t is transition matrix, w_t is process noise drawn from a zero mean multivariate normal distribution with covariance Q_t, and B_{t-1} is an optional control matrix applied to control vector u_{t-1}. The updated state estimate is based on the prediction and observation (measurement) according to the following equation:

$$\hat{s}_{t|t} = \hat{s}_{t|t-1} + K_t \cdot \left(z_t - H_t \hat{s}_{t|t-1}\right) \tag{7}$$

where K_t is optimal Kalman gain, expressed by Eq. (7), z_t is the measurement, and H_t is the observation model which maps the true state space into the observed space:

$$K_t = P_{t|t-1} H_t^{\mathrm{T}} S_t^{-1} \tag{8}$$

The variable $P_{t|t-1}$ is a prior estimate covariance and S_t is residual covariance, expressed by the equation:

$$S_t = H_t P_{t|t-1} H_t^{\mathrm{T}} + R_t \tag{9}$$

where R_t is the observation noise covariance.

In the presented system Kalman filtering was used to smooth the movement trajectory resulting in raising gesture recognition effectiveness and improving accuracy of writing / drawing on the whiteboard. The filtering was implemented using the OpenCV library [7].

The state of the system (hand) at the given moment is expressed by (x, y) position, vertical velocity and horizontal velocity according to Eq. (10);

$$s_t = [x_t, y_t, v_t^x, v_t^y] \tag{10}$$

The state in time t is related to state in time $t-1$ by a function of velocity and so the transposition matrix takes values as follows:

$$F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where dt, expressed by Eq. (11), is a time modification of the velocity and depends on the camera frame rate f_{FR} and the number of frames $n_{\tau_0}^{\tau_1}$ basing on which a singular motion vector is created [4]:

$$dt = c \cdot \frac{n_{\tau_0}^{\tau_1}}{f_{FR}} \tag{12}$$

Constant c, equals 10, resulting from the chosen velocity unit and is used to scale the velocity values. For obtained frame rate equal 15 FPR and the singular motion vector based on three successive frames dt equals 2. Thus, applying the transition matrix to state at time step $t-1$ results in predicted state as follows:

$$\hat{s}_{t|t-1} = \begin{bmatrix} x_{t|t-1} = x_{t-1|t-1} + 2 \cdot v_{t-1|t-1}^x \\ y_{t|t-1} = y_{t-1|t-1} + 2 \cdot v_{t-1|t-1}^y \\ v_{t|t-1}^x = v_{t-1|t-1}^x \\ v_{t|t-1}^y = v_{t-1|t-1}^y \end{bmatrix} \tag{13}$$

Measurement matrix H_t is initialized to identity, as well as the posterior error covariance $P_{t|t}$. The process noise covariance Q_t and the observation noise covariance R_t are set to diagonal matrices with values equal 10^{-5} and 10^{-1}, respectively.

A comparison of grouped gestures recognition efficacy without and with Kalman filters is presented in Tab. 1 and a visual assessment of the related results can be seen in Fig. 3.

Table 1. Comparison of grouped gesture recognition efficacy without and with Kalman filters [%] (for 20 gesture repetitions made by 20 people)

Gesture	Without Kalman filter	With Kalman filter
Full screen	91.19	90.99
Quitting full screen	91.78	86.57
Closing application	62.96	88.89

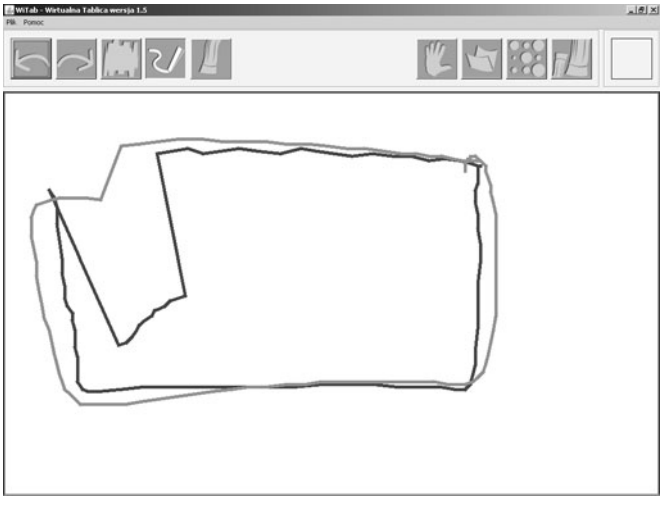


Fig. 3. Comparison of rectangular shapes created in poor light conditions without the Kalman filtering (darker line) and with the Kalman filtered hand position tracking (brighter line)

3.2 Fuzzy Logic Interpreter

Representing a gesture as a singular change of speed and direction over particular time interval often led to interpreting it as moving hand up – in the beginning phase of the movement – or as moving hand down – in the ending phase. Therefore, the movement trajectory in the second approach has been modeled by motion vectors created for points in time moments t_1 and t_2, in relation to the moments t_0 and t_1, respectively, as presented in Fig. 2 and gestures were analyzed considering a possibility of a local change of direction. Time intervals $t_1 - t_0$ and $t_2 - t_1$, expressed in the number of frames retrieved from a camera, depend on camera frame rate [4].

Fuzzy rules were created basing on speed and direction of motion vector over time interval $t_2 - t_1$ and $t_1 - t_0$ separately for left and right hand. Eight linguistic variables were proposed, i.e.: speed of left and right hand in time interval $t_2 - t_1$, speed of left and right hand in time interval $t_1 - t_0$, direction of left and right hand

in time interval t_2-t_1, direction of left and right hand in time interval t_1-t_0, denoted as v_{21}^L, v_{21}^R, v_{10}^L, v_{10}^R, d_{21}^L, d_{21}^R, d_{10}^L, d_{10}^R, respectively. Four linguistic terms were used for speed, i.e.: *very small*, *small* (denoted later as *vsmall*), *medium and high*, represented by triangular functions as shown in Fig. 4a. Fuzzy sets were identical for all four variables. For directions the terms used were *north, east, south* and *west* and fuzzy sets were also formed using triangular functions as shown in Fig. 4b.

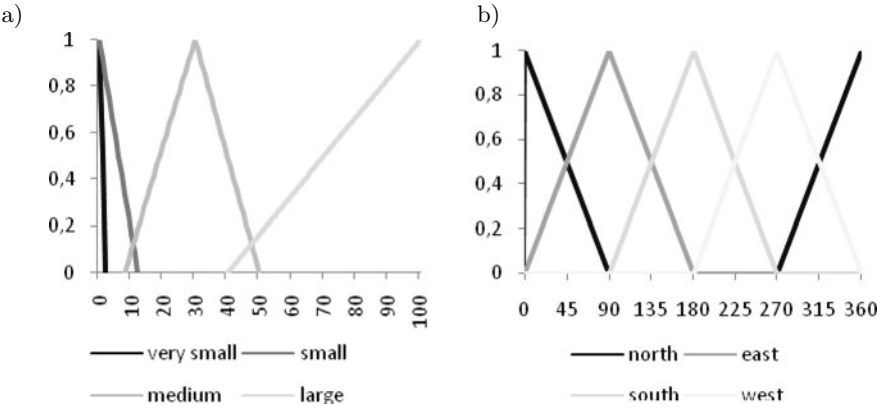


Fig. 4. Fuzzy sets for linguistic variables speed (a) and direction (b)

The zero-order Takagi-Sugeno inference model which bases on singletons was used to express discrete rule outputs representing gesture classes. The output of the system was the maximum of all rule outputs. When this value was lower than 0.5 a movement was labeled as *no gesture*. This enabled to efficiently solve the problem of classifying meaningless transitions between each two gestures to one of the gesture classes. The total number of rules equaled 30. Two examples of rules expressed in FCL code are given below [4]:

```
// beginning phase of hand movement in the left direction (for semi-circular motion) for left hand
RULE 1: IF directionLt0 IS north AND directionLt1 IS west AND velocityLt0 IS NOT small AND velocityLt1 IS NOT small AND velocityRt0 IS vsmall AND velocityRt1 IS vsmall THEN gesture IS g1;
// rotate left
RULE 29: IF directionLt0 IS south AND directionLt1 IS south AND directionRt0 IS north AND directionRt1 IS north AND (velocityLt1 IS NOT vsmall AND velocityLt0 IS NOT vsmall) AND (velocityRt1 IS NOT vsmall AND velocityRt0 IS NOT vsmall) THEN gesture IS g7;
```

The first rule describes the beginning phase of semi-circular left hand movement from right to the left side. Therefore, d_{10}^R is north and d_{21}^R is west. Since the gesture involves left hand only, the speed of the right hand should be very

small. If the right hand is not present in an image, 0.0 values are given as an input to the fuzzy inference system for variables v_{21}^R and v_{10}^R. The second rule represents the gesture associated with rotating the displayed object. During the gesture performing, the left hand moves down and the right hand moves up. No local change of direction is allowed. For this reason, both d_{21}^L and d_{10}^L are south and d_{21}^R, d_{10}^R are north. While making gestures involving both hands, speed of each hand movement can be lower than when performing a single hand gesture. Therefore, contrary to the first rule the second one allows for small speed.

Again 20 persons took part in tests. Each person was asked to repeat each gesture 18 times. Among these 18 repetitions 10 middle gesture representations were chosen. Since the system analyzes motion vectors for time intervals t_2-t_1 and t_1-t_0 in relation to each obtained camera frame, among each gesture representation there were many assignments to the particular gesture class. Sample results of a comparison between fuzzy rule-based recognition and recognition based on fixed thresholds with the analysis of global motion vector change are presented in Tab. 2.

Table 2. Gesture recognition effectiveness for the system employing fuzzy inference and without a module of fuzzy inference, for one hand gestures [%]

	With fuzzy logic						No fuzzy logic					
	Left	Right	Up	Down	Hand steady	No gesture	Left	Right	Up	Down	Hand steady	No gesture
Left	95.0	0.0	2.3	2.6	0.0	0.1	89.5	0.0	4.9	5.6	0.0	0.0
Right	0.0	94.2	2.9	2.7	0.0	0.2	0.0	89.6	5.8	4.6	0.0	0.0
Up	0.9	0.5	98.6	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
Down	2.2	0.9	0.0	96.9	0.0	0.0	0.0	0.0	0.0	99.8	0.0	0.2
Hand steady	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	73.3	16.7

4 Audio Modality: Speech Stretcher

A non-uniform real-time scale speech modification algorithm, was designed to improve the perception of speech by people with the hearing resolution deficit [8]. The software employing this algorithm enables to use an ultra-portable computer (e.g. smartphone) as a speech communication interface for people suffering from certain type of central nervous system impairments, which can impede learning.

The block diagram of the proposed algorithm is presented in Fig. 5. The algorithm provides a combination of voice activity detection, vowel detection, rate of speech estimation and time-scale modification algorithms. Signal processing is performed in time frames in the following order:

1. voice activity detector examines speech presence,
2. for noisy components the frame synchronization procedure is performed; if the output signal is not synchronized with the input then noise sample frames are not sent to the output,
3. speech sample frames are tested in order to find vowels,
4. information about vowels locations is used by the rate of speech estimator to determine the speech rate,
5. speech frames are stretched up with different stretching factors.

As speech signal is usually unrepeatable and often modeled as a stochastic process, above operations (3), (4) and (5) demand a heuristic approach to computing.

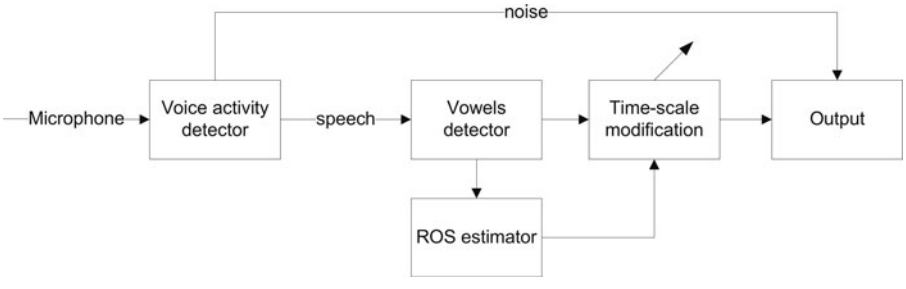


Fig. 5. Non-uniform real-time scale speech modification algorithm block diagram

The vowel detection algorithm is based on the assumption that all vowels amplitude spectra are consistent. To quantify this similarity a parameter called PVD (peak-valley difference) was used [9]. Initially PVD was introduced for the robust voice activity detection. It is defined by the following formula (Eq. (14)):

$$PVD(VM, A) = \frac{\sum_{k=0}^{N-1} (A(k) \cdot VM(k))}{\sum_{k=0}^{N-1} VM(k)} - \frac{\sum_{k=0}^{N-1} (A(k) \cdot (1 - VM(k)))}{\sum_{k=0}^{N-1} (1 - VM(k))} \tag{14}$$

where $PVD(VM, A)$ is the value of peak-valley difference for one frame of the input signal, $A(k)$ is the value of the kth spectral line of the input signal magnitude spectrum and $VM(k)$ is the value of the kth value in the vowel model vector.

The VM is created in the training stage on the basis of the average magnitude spectra calculated for the pre-recorded vowels. The model consists of binary values, where 1 is placed in the position of the peak in the average magnitude spectrum and 0 for all other positions. When the magnitude spectrum of the input signal is highly correlated with the vowels spectra, the PVD value is high. Therefore, the PVD have higher values for the vowels than for consonants or silence parts.

Vowels detection is executed only for speech frames. The algorithm is based on time frames with the duration of 23 ms. Each signal frame is windowed using triangular window defined as:

$$\omega(n) = \begin{cases} \dfrac{2n}{L}, & 1 \le n \le \dfrac{L+1}{2} \\ \dfrac{2(L-n+1)}{L}, & \dfrac{L}{2}+1 \le n \le L \end{cases} \tag{15}$$

where L is the size of the window and n is the sample number. This type of window ensures a higher accuracy of vowel detection than other shapes.

Vowel detection requires the initialization step which is performed in parallel to the initialization of the voice activity detection algorithm. In this step the threshold for the PVD is calculated as the mean value of first 40 frames of the signal according to the formula:

$$Pth = C\frac{\sum_{n=1}^{N} PVD(n)}{N} \tag{16}$$

where Pth is initial value of the threshold, $PVD(n)$ is the value of peak-valley difference for the nth signal frame, N is the number of frames that were used for initial threshold calculation, C is the correction factor. The correction factor was selected experimentally and was set to 1.1.

For every signal frame the PVD value is determined and smoothed by calculating the average of the last three values. The signal frame is marked as a vowel when: the value of the smoothed PVD is higher than Pth threshold and it has a local maximum in the PVD curve or its value is higher than 70% of the value of the last local maximum. If the value is lower than Pth, then the decision of the voice activity detector is corrected and the frame is marked as silence. For other situations the frame is assigned to the consonant class.

Rate of Speech (ROS) is a useful parameter in many speech processing systems. For the most part it is used in the automatic speech recognition (ASR). In the ASR many speech parameters are highly related to ROS. Hence, ROS is used to adjust the HMM model for different speech rates [10].

For real-time unknown input signal, ROS estimation could be done only by statistical analysis. In this work, as ROS definition, the VPS parameter is used, as the derivate of SPS measure. Therefore, ROS is defined as (Eq. (17)):

$$\mathrm{ROS}(n) = \frac{N_{vowels}}{\Delta t} \tag{17}$$

Mean value and standard deviation of ROS calculated for the different speech rates for 3 persons reading the same phrase with three speech rates: high, medium and low are shown in Tab. 3.

It can be seen that, because of the high value of the standard deviation (nearly 0.6 for all classes) and as a consequence of the low distance between the neighbor classes, only two classes could be separated linearly using the instantaneous ROS

Table 3. Mean value and standard deviation of ROS calculated for different speech rates

speech rate	low	medium	high
μ(ROS) [vowels/s]	2.23	2.4	2.56
Δ(ROS) [vowels/s]	0.6	0.58	0.57

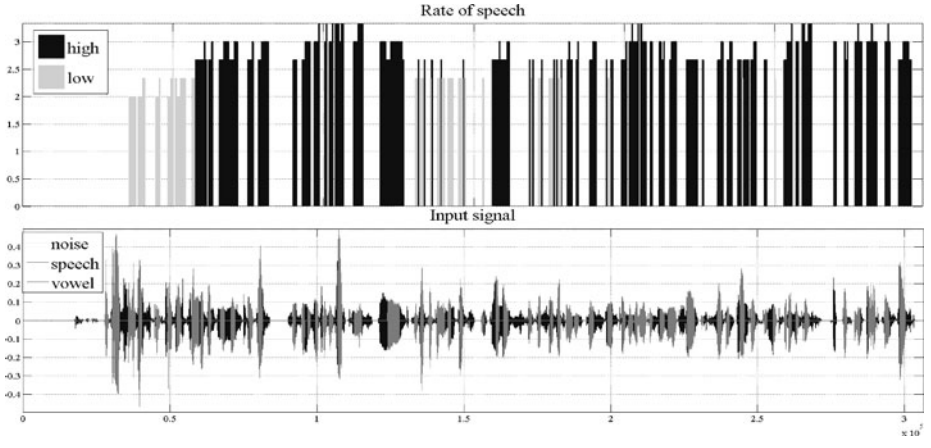


Fig. 6. Speech rate recognition for high speech rate female speech

value. On the basis of the statistics, *ROSth* value was set to 2.5 vowel/s. In Fig. 6 waveforms corresponding to the recorded female high rate speech with estimated speech rate are presented.

For time-scale modification of speech an algorithm based on the SOLA algorithm (Synchronous Overlap-and-Add) was applied which in the fundamental form uses constant values of the analysis/synthesis frame sizes and analysis/synthesis time shift [11] as well ensures quality of the processed speech nearly as good as for the other methods [12, 13].

To achieve high quality of the stretched speech, analysis/synthesis frame size and analysis time shift should be selected properly i.e. frame length L should cover at least one period of the lowest speech component and in the synthesis stage, for all used scaling factors $\alpha(t)$, the overlap size should be at least $L/3$ length. For the designed algorithm L value was set to 46 ms and the analysis time shift Sa to 11.5 ms.

The synthesis time shift Ss is dependent on the current value of the scaling factor $\alpha(t)$. The scaling factor is defined as in Eq. (18):

$$\alpha(t) = \frac{Ss}{Sa} \tag{18}$$

Synchronization between two synthesized overlapped frames is obtained by calculating the highest similarity point which is determined by the maximum of the cross-correlation function calculated for the overlapped parts of successive frames.

To reduce the duration of the stretched speech and to improve quality of the modified signal, the scaling factor is changed accordingly to different speech content. For both speech rates (low and high) vowels are stretched up with the designed scale factor value ($\alpha(t) = \alpha_d$, being the value that is specified for the processing), and noise is not modified ($\alpha(t) = 1$) or removed from the signal dependently on the input/output synchronization state. For the low rate speech consonants are stretched up with the factor lower than α_d and equal to $\alpha(t) = 0.8 \cdot \alpha_d$, and for the high rate speech consonants are not stretched ($\alpha(t) = 1$).

Quality of the whole speech stretching algorithm was assessed in subjective tests performed for 19 healthy persons (2 women, 17 men) [8]. Each person had to assess quality of speech stretched using the typical SOLA algorithm implementation and the proposed algorithm. Two values of the stretching factors were chosen: 1.9 and 2.1. Four recordings were used during the experiment: two spoken with the low rate, and two with the high rate. Both of them were spoken by a woman and a man. In all recordings the same phrase was uttered.

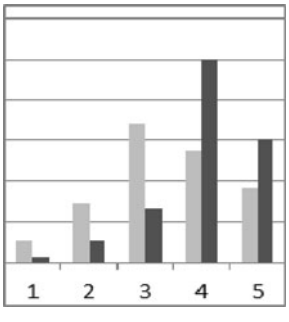


Fig. 7. Processed speech quality assessment for high speech rate (grey bars represent SOLA algorithm, darker bars represent the proposed heuristic algorithm)

Three parameters were rated during tests: signal quality, speech naturalness and speech ineligibility. The assessment was made using the following scale: 1 – very poor, 2 – poor, 3 – medium, 4 – good, 5 – very good. Test results revealed that for both speech rates, as well as for all parameter values, histograms that represent the proposed algorithm assessment have higher placed gravity centers than for the SOLA algorithm. As is seen in Fig. 7 for the high rate speech this difference becomes more significant.

Recently an implementation of the algorithm working in real-time have been performed on the mobile device (the Apple iPhone platform).

5 Conclusions

Authors of this paper and their co-workers believe that in the future multimodal interfaces will enable a more natural control of computers with speech, gestures, eye movements and face expression engaging human senses interactively in a much broader way than today. Consequently, future learning systems will engage:

- blending technologies (personal and institutional),
- online and onsite convergence,
- customized pedagogy,
- students as knowledge generators, not just consumers,
- immersive, gaming environment for teaching.

Besides three examples presented in this paper, many more multimodal interfaces are currently under development at our Multimedia Systems Department, including: biofeedback-based brain hemispheric synchronizing man-machine interface [14], virtual computer touch pad [15], browser controller employing head movements [16], intelligent tablet pen [17] and scent emitting computer interface [18].

Acknowledgements

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

Reference

1. Xu, R., Da, Y.: A Computer Vision based Whiteboard Capture System. In: IEEE Workshop on Applications of Computer Vision, WACV 2008, pp. 1–6 (2008)
2. Maes, P., Mistry, P.: Unveiling the "Sixth Sense", game-changing wearable tech. In: TED 2009, Long Beach, CA, USA (2009)
3. Mistry, P., Maes, P.: SixthSense – A Wearable Gestural Interface. In: SIGGRAPH Asia 2009, Emerging Technologies.Yokohama, Japan (2009)
4. Lech, M., Kostek, B.: Fuzzy Rule-based Dynamic Gesture Recognition Employing Camera & Multimedia Projector. In: Advances in Intelligent and Soft Computing: Multimedia & Network Information System. Springer, Heidelberg (2010)
5. Lech, M., Kostek, B.: Gesture-based Computer Control System Applied to the Interactive Whiteboard. In: 2nd International Conference on Information Technology ICIT 2010, Gdansk, June 28-30, pp. 75–78 (2010)
6. Kalman, R.R.: A New Approach to Linear Filtering and Prediction Problems. Transaction of the ASME – Journal of Basic Engineering, 35–45 (1960)
7. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, Sebastopol (2008)

8. Kupryjanow, A., Czyzewski, A.: Real-time speech-rate modification experiments. Audio Engineering Society Convention, preprint No. 8052, London, GB, May 22-25 (2010)
9. Moattar, M., Homayounpour, M., Kalantari, N.: A new approach for robust real-time voice activity detection using spectral pattern. In: ICASSP Conference, March 14-19 (2010)
10. Zheng, J., Franco, H., Weng, F., Sankar, A., Bratt, H.: Word-level rate-of-speech modeling using rate-specific phones and pronunciations. In: Proc. IEEE Int. Conf. Acoust. Speech Signal Process, Istanbul, vol. 3, pp. 1775–1778 (2000)
11. Pesce, F.: Realtime-Stretching of Speech Signals. In: Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX 2000), Verona, Italy, December 7-9 (2000)
12. Verhelst, W., Roelands, M.: An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In: ICASSP 1993, Minneapolis, USA, April 27-April 30, vol. 2 (1993)
13. Kupryjanow, A., Czyzewski, A.: Time-scale modification of speech signals for supporting hearing impaired schoolchildren. In: Proc. of International Conference NTAV/SPA, New Trends in Audio and Video, Signal Processing: Algorithms, Architectures, Arrangements and Applications, Poznań, September 24-25 , pp. 159–162 (2009)
14. Kaszuba, K., Kopaczewski, K., Odya, P., Kostek, B.: Biofeedback-based brain hemispheric synchronizing employing man-machine interface. In: Tsihrintzis, G.A., et al. (eds.) The 3rd International Symposium on Intelligent and Interactive Multimedia: Systems and Services KES 2010, Baltimore, USA, July 28-30, pp. 2010–2030. Springer, Heidelberg (2010)
15. Kupryjanow, A., Kunka, B., Kostek, B.: UPDRS tests for Diagnosis of Parkinson's Disease Employing Virtual-Touchpad. In: 4th International Workshop on Management and Interaction with Multimodal Information Content – MIMIC 2010, Bilbao, Spain, August 30-September 3 (2010)
16. Kosikowski, L., Czyzewski, A., Dalka, P.: Multimedia Browser Controlled by Head Movements. In: 37 Conf. and Exhibition on Computer Graphics and Interactive Techniques, SIGGRAPH, Los Angeles, USA, July 25-29 (2010)
17. Odya, P., Czyzewski, A., Grabkowska, A., Grabkowski, M.: Smart Pen – new multimodal computer control tool for graphomotorical therapy. Intelligent Decision Technologies Journal 4(3), 197–209 (2010)
18. Smulko, J., Kotarski, M., Czyzewski, A.: Fluctuation-enhanced scent sensing using a single gas sensor. Sensors & Actuators: B. Chemical (to be printed in 2011)

Relational Mining in Spatial Domains: Accomplishments and Challenges

Donato Malerba, Michelangelo Ceci, and Annalisa Appice

Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro
via Orabona, 4 - 70126 Bari - Italy
{malerba,ceci,appice}@di.uniba.it

Abstract. The rapid growth in the amount of spatial data available in Geographical Information Systems has given rise to substantial demand of data mining tools which can help uncover interesting spatial patterns. We advocate the relational mining approach to spatial domains, due to both various forms of spatial correlation which characterize these domains and the need to handle spatial relationships in a systematic way. We present some major achievements in this research direction and point out some open problems.

1 Introduction

Several real world applications, such as fleet management, environmental and ecological modeling, remote sensing, are the source of a huge amount of spatial data, which are stored in spatial databases of Geographic Information Systems (GISs). A GIS is a software system that provides the infrastructure for editing, storing, analyzing and displaying spatial objects [10]. Popular GISs (e.g. ArcView, MapInfo and Open GIS) have been designed as a toolbox that allows planners to explore spatial data by zooming, overlaying, and thematic map coloring. They are provided with functionalities that make the spatial visualization of individual variables effective, but overlook complex multi-variate dependencies.

The solution to this limitation is to integrate GIS with *spatial data mining* tools [25]. Spatial data mining investigates how interesting, but not explicitly available, knowledge (or pattern) can be extracted from spatial data [34]. Several algorithms of spatial data mining have been reported in the literature for both predictive tasks (e.g., regression [24], [14], and localization [32]) and descriptive tasks (e.g., clustering [28,16] and discovery of association rules [19,3], co-location patterns [33], subgroups [18], emerging patterns [6], spatial trends [11] and outliers [31]).

Spatial data mining differs from traditional data mining in two important respects. First, spatial objects have a locational property which implicitly defines several spatial relationships between objects such as topological relationships (e.g., intersection, adjacency), distance relationships, directional relationships (e.g., north-of), and hybrid relationships (e.g., parallel-to). Second, attributes of spatially interacting (i.e., related) units tend to be statistically correlated. *Spatial*

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 16–24, 2011.

cross-correlation refers to the correlation between two distinct attributes across space (e.g., the employment rate in a city depends on the business activities both in that city and in its neighborhood). *Autocorrelation* refers to the correlation of an an attribute with itself across space (e.g., the price level for a good at a retail outlet in a city depends on the price for the same good in the nearby). In geography, spatial autocorrelation is justified by Tobler's First law of geography, according to which "everything is related to everything else, but near things are more related than distant things" [35].

The network of data defined by implicit spatial relationships can sometime reveal the network of statistical dependencies in a spatial domain (e.g., region adjacency is a necessary condition for autocorrelation of air pollution). Nevertheless, the two concepts do not necessarily coincide. On the contrary, it is the identification of some form of spatial correlation which help to clarify what are the relevant spatial relationships among the infinitely many that are implicitly defined by locational properties of spatial objects.

The presence of spatial dependence is a clear indication of a violation of one of the fundamental assumptions of classic data mining algorithms, that is, the independent generation of data samples. As observed by LeSage and Pace [21], "anyone seriously interested in prediction when the sample data exhibit spatial dependence should consider a spatial model", since this can take into account different forms of spatial correlation. In addition to predictive data mining tasks, this consideration can also be applied to descriptive tasks, such as spatial clustering or spatial association rule discovery. The inappropriate treatment of sample data with spatial dependence could obfuscate important insights and observed patterns may even be inverted when spatial autocorrelation is ignored [20].

In order to accommodate several forms of spatial correlation, various models have been developed in the area of spatial statistics. The most renowned types of models are the spatial lag model, the spatial error model, and the spatial cross-regressive model [1], which consider autocorrelation, correlation of errors, and cross-correlation, respectively.

Despite the many successful applications of these models, there are still several limitations which prevent their wider usage in a spatial data mining context. First, they require the careful definition of a spatial weight matrix in order to specify to what extent a spatially close observation in a given location can affect the response observed in another location. Second, there is no clear method on how to express the contribution of different spatial relationships (e.g., topological and directional) in a spatial weight matrix. Third, spatial relationships are all extracted in a pre-processing step, which typically ignores the subsequent data mining step. In principle, a data mining method, which can check whether a spatial relationship contributes to defining a spatial dependency, presents the advantage of considering only those relationships that are really relevant to the task at hand. Fourth, all spatial objects involved in a spatial phenomena are uniformly represented by the same set of attributes. This can be a problem when spatial objects are heterogeneous (e.g., city and roads). Fifth, there is no clear distinction between the *reference* (or target) *objects*, which are the main

subject of analysis, and the *task-relevant objects*, which are spatial objects "in the neighborhood" that can help to account for the spatial variation.

A solution to above problems is offered by latest developments in *relational mining* or *relational learning*. Indeed, relational mining algorithms can be directly applied to various representations of networked data, i.e. collections of interconnected entities. By looking at spatial databases as a kind o networked data where entities are spatial objects and connections are spatial relationships, the application of relational mining techniques appears straightforward, at least in principle. Relational mining techniques can take into account the various forms of correlation which bias learning in spatial domains. Furthermore, discovered relational patterns reveal those spatial relationships which correspond to spatial dependencies.

This relational mining approach to spatial domains has been advocated in several research papers [18,23,12]. Major accomplishments in this direction have been performed, but there are still many open problems which challenges researchers. In the rest of the paper, we pinpoint the important issues that need to be addressed in spatial data mining, as well as the opportunities in this emerging research direction.

2 Integration with Spatial Databases

Spatial data are stored in a set of *layers*, that is, database relations each of which has a number of elementary attributes, called thematic data, and a geometry attribute represented by a vector of coordinates. The computation of spatial relationships, which are fundamental for querying spatial data, is based on spatial joins [30]. To support the efficient computation of spatial joins, special purpose indexes like Quadtrees and Kd-tree [27] are used.

Integration can be tight, as in SubgroupMiner [18] and Mrs-SMOTI [24], or loose as in ARES [2]. A tight integration:

- guarantees the applicability of spatial data mining algorithms to large spatial datasets;
- exploits useful knowledge of spatial data model available, free of charge, in the spatial database;
- avoids useless preprocessing to compute spatial relationships which do not express statistical dependencies.

A loose integration is less efficient, since it uses a middle layer module to extract both spatial attributes and relationships independently of the specific data mining step. On the other hand, this decoupling between the spatial database and the data mining algorithm allows researchers to focus on general aspects of the relational data mining task, and to exploit important theoretical and empirical results. A systematic study of these integration approaches should lead to valuable information on how a spatial data mining task should be methodologically dealt with.

Many relational mining methods take advantage of knowledge on the data model (e.g., foreign keys), which is obtained free of charge from the database schema, in order to guide the search process. However, this approach does not suit spatial databases, since the database navigation is also based on the spatial relationships, which are not explicitly modeled in the schema. The high number of spatial relationships is a further complication, since each individual relationship can become insignificant on its own, requiring the use of some form of spatial aggregation [12].

3 Dealing with Hierarchical Representations of Objects and Relationships

Both spatial objects and spatial relationships are often organized in taxonomies typically represented by hierarchies [37]. By descending/ascending through a hierarchy it is possible to view the same spatial object/relationship at different levels of abstraction (or granularity). Spatial patterns involving the most abstract spatial objects and relationships can be well supported but at the same time they are the less confident. Therefore, spatial data mining methods should be able to explore the search space at different granularity levels in order to find the most interesting patterns (e.g., the most supported and confident). In the case of granularity levels defined by a containment relationship (e.g., Bari → Apulia → Italy), this corresponds to exploring both global and local aspects of the underlying phenomenon. Geo-associator [19] and SPADA [22] are two prominent examples of spatial data mining systems which automatically support this multiple-level analysis. However, there is still no clear methodization for extracting, representing and exploiting hierarchies of spatial objects and relationships in knowledge discovery.

4 Dealing with Spatial Autocorrelation

Relational mining algorithms exploit two sources of correlation when they discover relational patterns: *local correlation*, i.e., correlation between attributes of each unit of analysis, and *within-network correlation*, i.e., correlation between attributes of the various units of analysis. In this sense, they are appropriate for spatial domains, which present both forms of correlation. For instance, the spatial subgroup mining system SubgroupMiner [18] is built on previous work on relational subgroup discovery [38], although it also allows numeric target variables, and aggregations based on (spatial) links. The learning system Mrs-SMOTI [24], which learns a tree-based regression model from spatial data, extends the relational system Mr-SMOTI [4] by associating spatial queries to nodes of model trees. UnMASC [12] is based on both the idea of the sequential covering algorithm developed in the relational data mining system CrossMine [39] and on aggregation-based methods originally proposed for relational classification [13].

However, predictive modeling in spatial domains still challenges most relational mining algorithms when autocorrelation on the target (or response)

variable is captured. Indeed, values of the target variable of unclassified units of analysis have to be inferred collectively, and not independently as most relational mining algorithm do. *Collective inference* refers to the simultaneous judgments regarding the values of response variables for multiple linked entities for which some attribute values are not known. Several collective inference methods (e.g., Gibbs sampling, relaxation labeling, and iterative classification) have been investigated in the context of relational learning. For the specific task of classification it has been proven that collective inference outperforms independent classification when the autocorrelation between linked instances in the data graph is high [17]. Collective inference in the context of spatial predictive modeling is still a largely unexplored area of research.

5 Dealing with Unlabeled Data

Learning algorithms designed for mining spatial data may require large sets of labeled data. However, the common situation is that only few labeled training data are available since manual annotation of the many objects in a map is very demanding. Therefore, it is important to exploit the large amount of information potentially conveyed by unlabeled data to better estimate the data distribution and to build more accurate classification models. To deal with this issue, two learning settings have been proposed in the literature: the semi-supervised setting and the transductive setting [29]. The former is a type of inductive learning, since the learned function is used to make predictions on any possible example. The latter asks for less - it is only interested in making predictions for the given set of unlabeled data.

Transduction [36] seems to be the most suitable setting for spatial classification tasks, for at least two reasons. First, in spatial domains observations to be classified are already known in advance: they are spatial objects on maps already available in a GIS. Second, transduction is based on a (semi-supervised) smoothness assumption according to which if two points x_1 and x_2 in a high-density region are close, then the corresponding outputs y_1 and y_2 should also be close [8]. In spatial domains, where closeness of points corresponds to some spatial distance measure, this assumption is implied by (positive) spatial autocorrelation. Therefore, we expect that a strong spatial autocorrelation should counterbalance the lack of labeled data, when transductive relational learners are applied to spatial domains. Recent results for spatial classification [7] and spatial regression tasks [5] give support to this expectation. Nevertheless, more experiments are needed to substantiate this claim.

6 Dealing with Dynamic Spatial Networks

Most of works on spatial data mining assume that the spatial structure is static. Nevertheless, changes may occur in many real-world applications (e.g., the public transport network can change). This causes the appearance and disappearance of spatial objects and spatial relationships over time, while properties of the spatial

objects may evolve. By analyzing these changes, we can follow variations, adapt tools and services to new demands, as well as capture and delay undesirable alterations. Moreover, time associated to changes represent a valuable source of information which should be modeled to better understand both the whole dynamics and each change in the course of dynamics.

In the literature, the task of change mining has been mainly explored for time-series, transactional data and tabular data, by focusing on the detection of significant deviations in the values of the attributes describing the data. However, detecting and analyzing changes on spatially referenced data is critical for many applications. For instance, by taking snapshots of over time of the spatial distribution of plant species, it is possible to monitor significant changes, which may reveal important ecological phenomena. Pekerskaya et al. [26] address the problem of mining changing regions by directly comparing models (cluster-embedded decision trees) built on the original data snapshots. This approach is suitable when there are data access constraints such as privacy concerns and limited data online availability. Ciampi et al. [9] consider the case of distributed streams of unidimensional numeric data, where each data source is a geo-referenced remote sensor which periodically records measures for a specific numeric theme (e.g., temperature, humidity). A combination of stream and spatial data mining techniques is used to mine a new kind of spatio-temporal patterns, called trend clusters, which are spatial clusters of sources for which the same temporal variation is observed over a time window.

Spatial networks demand for attention not only on the attributes which may describe nodes and links but also on the structural and topological aspects of the network, namely the relationships among the nodes and the kind of links which connect the nodes. In this direction, research on network analysis has mainly investigated graph-theoretical approaches which oversimplify the representation of spatial networks. Indeed, graph-theory mainly investigates structural aspects, such as distance and connectivity, in homogeneous networks, while it almost ignores the data heterogeneity issue, which is typical of spatial networks, where nodes are of different types (e.g. in public transport networks, public services and private houses should be described by different feature sets), and relationships among nodes can be of different nature (e.g. connection by bus, railway or road). Methods for learning and inference with networks of heterogeneous data have been investigated in the context of statistical relational learning [15], however the scalability issue that characterizes many most statistical relational learning methods makes their application very challenging in the case of dynamic networks due to continuous changes in the network.

7 Conclusions

In this paper, we have advocated a relational mining approach to spatial domains, and we have presented some major research achievements in this direction. Research results are encouraging but there are still many open problems which challenge current relational mining systems, namely:

1. a methodological support to the integration of spatial database technology with data mining tools;
2. the potentially infinitely many spatial relationships which are implicitly defined by spatial objects;
3. the efficient discovery of spatial patterns at various levels of granularity;
4. the demand for collective inference in predictive models which capture autocorrelation;
5. the exploitation of the many unlabeled spatial objects in a semi-supervised or transductive setting;
6. the need of new types of patters which capture the interactions between both spatial and temporal dimensions in spatially static structures;
7. the structural changes of dynamic networks with heterogeneous spatial objects.

Obviously, this list of challenges is not exhaustive, but rather it is indicative of the necessity for developing synergies between researchers interested in spatial data mining and relational learning. Some efforts have been made, but the two research communities still work in relative isolation from one another, with little methodological common ground. Nonetheless, there is good cause for optimism: there are many real-world applications which cry out for this collaboration.

References

1. Anselin, L., Bera, A.: Spatial dependence in linear regression models with an application to spatial econometrics. In: Ullah, A., Giles, D. (eds.) Handbook of Applied Economics Statistics, pp. 21–74. Springer, Heidelberg (1998)
2. Appice, A., Berardi, M., Ceci, M., Malerba, D.: Mining and filtering multi-level spatial association rules with ARES. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 342–353. Springer, Heidelberg (2005)
3. Appice, A., Ceci, M., Lanza, A., Lisi, F.A., Malerba, D.: Discovery of spatial association rules in georeferenced census data: A relational mining approach. Intelligent Data Analysis 7(6), 541–566 (2003)
4. Apice, A., Ceci, M., Malerba, D.: Mining model trees: A multi-relational approach. In: Horváth, T., Yamamoto, A. (eds.) ILP 2003. LNCS (LNAI), vol. 2835, pp. 4–21. Springer, Heidelberg (2003)
5. Appice, A., Ceci, M., Malerba, D.: Transductive learning for spatial regression with co-training. In: Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C.-C. (eds.) SAC, pp. 1065–1070. ACM, New York (2010)
6. Ceci, M., Appice, A., Malerba, D.: Discovering emerging patterns in spatial databases: A multi-relational approach. In: Kok, J.N., Koronacki, J., de Mántaras, R.L., Matwin, S., Mladenic, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 390–397. Springer, Heidelberg (2007)
7. Ceci, M., Appice, A., Malerba, D.: Transductive learning for spatial data classification. In: Koronacki, J., Ras, Z.W., Wierzchon, S.T., Kacprzyk, J. (eds.) Advances in Machine Learning I. Studies in Computational Intelligence, vol. 262, pp. 189–207. Springer, Heidelberg (2010)

8. Chapelle, O., Schölkopf, B.B., Zien, A.: Semi-supervised learning. MIT Press, Cambridge (2006)
9. Ciampi, A., Appice, A., Malerba, D.: Summarization for geographically distributed data streams. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6278, pp. 339–348. Springer, Heidelberg (2010)
10. Densham, P.: Spatial decision support systems. Geographical Information Systems: Principles and Applications, 403–412 (1991)
11. Ester, M., Gundlach, S., Kriegel, H., Sander, J.: Database primitives for spatial data mining. In: Proceedings of the International Conference on Database in Office, Engineering and Science, BTW 1999, Freiburg, Germany (1999)
12. Frank, R., Ester, M., Knobbe, A.J.: A multi-relational approach to spatial classification. In: Elder IV, J.F., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) KDD, pp. 309–318. ACM, New York (2009)
13. Frank, R., Moser, F., Ester, M.: A method for multi-relational classification using single and multi-feature aggregation functions. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 430–437. Springer, Heidelberg (2007)
14. Gao, X., Asami, Y., Chung, C.: An empirical evaluation of spatial regression models. Computers & Geosciences 32(8), 1040–1051 (2006)
15. Getoor, L., Taskar, B. (eds.): Introduction to Statistical Relational Learning. MIT Press, Cambridge (2007)
16. Han, J., Kamber, M., Tung, A.K.H.: Spatial Clustering Methods in Data Mining: A Survey. In: Geographic Data Mining and Knowledge Discovery, pp. 1–29. Taylor and Francis, Abington (2001)
17. Jensen, D., Neville, J., Gallagher, B.: Why collective inference improves relational classification. In: Kim, W., Kohavi, R., Gehrke, J., DuMouchel, W. (eds.) KDD, pp. 593–598. ACM, New York (2004)
18. Klösgen, W., May, M.: Spatial subgroup mining integrated in an object-relational spatial database. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, pp. 275–286. Springer, Heidelberg (2002)
19. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: Egenhofer, M.J., Herring, J.R. (eds.) SSD 1995. LNCS, vol. 951, pp. 47–66. Springer, Heidelberg (1995)
20. Kühn, I.: Incorporating spatial autocorrelation invert observed patterns. Diversity and Distributions 13(1), 66–69 (2007)
21. LeSage, J.P., Pace, K.: Spatial dependence in data mining. In: Grossman, R., Kamath, C., Kegelmeyer, P., Kumar, V., Namburu, R. (eds.) Data Mining for Scientific and Engineering Applications, pp. 439–460. Kluwer Academic Publishing, Dordrecht (2001)
22. Lisi, F.A., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning 55, 175–210 (2004)
23. Malerba, D.: A relational perspective on spatial data mining. IJDMMM 1(1), 103–118 (2008)
24. Malerba, D., Ceci, M., Appice, A.: Mining model trees from spatial data. In: Jorge, A., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 169–180. Springer, Heidelberg (2005)
25. Malerba, D., Esposito, F., Lanza, A., Lisi, F.A., Appice, A.: Empowering a GIS with inductive learning capabilities: The case of INGENS. Journal of Computers, Environment and Urban Systems 27, 265–281 (2003)

26. Pekerskaya, I., Pei, J., Wang, K.: Mining changing regions from access-constrained snapshots: a cluster-embedded decision tree approach. Journal of Intelligent Information Systems 27(3), 215–242 (2006)
27. Samet, H.: Applications of spatial data structures. Addison-Wesley, Longman (1990)
28. Sander, J., Ester, M., Kriegel, H., Xu, X.: Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. Data Mining and Knowledge Discovery 2(2), 169–194 (1998)
29. Seeger, M.: Learning with labeled and unlabeled data. Technical report, University of Edinburgh (2001)
30. Shekhar, S., Chawla, S.: Spatial databases: A tour. Prentice Hall, Upper Saddle River (2003)
31. Shekhar, S., Huang, Y., Wu, W., Lu, C.: What's spatial about spatial data mining: Three case studies. In: Grossman, R., Kamath, C., Kegelmeyer, P., Kumar, V., Namburu, R. (eds.) Data Mining for Scientific and Engineering Applications. Massive Computing, vol. 2, pp. 357–380. Springer, Heidelberg (2001)
32. Shekhar, S., Schrater, P.R., Vatsavai, R.R., Wu, W., Chawla, S.: Spatial contextual classification and prediction models for mining geospatial data. IEEE Transactions on Multimedia 4(2), 174–188 (2002)
33. Shekhar, S., Vatsavai, R., Chawla, S.: Spatial classification and prediction models for geospatial data mining. In: Miller, H., Han, J. (eds.) Geographic Data Mining and Knowledge Discovery, 2nd edn., pp. 117–147. Taylor & Francis, Abington (2009)
34. Shekhar, S., Zhang, P., Huang, Y.: Spatial data mining. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 837–854. Springer, Heidelberg (2010)
35. Tobler, W.: A computer movie simulating urban growth in the detroit region. Economic Geography 46, 234–240 (1970)
36. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
37. Vert, G., Alkhaldi, R., Nasser, S., Harris Jr., F.C., Dascalu, S.M.: A taxonomic model supporting high performance spatial-temporal queries in spatial databases. In: Proceedings of High Performance Computing Systems (HPCS 2007), pp. 810–816 (2007)
38. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg (1997)
39. Yin, X., Han, J., Yang, J., Yu, P.S.: CrossMine: Efficient classification across multiple database relations. In: ICDE, pp. 399–411. IEEE Computer Society, Los Alamitos (2004)

Towards Programming Languages for Machine Learning and Data Mining (Extended Abstract)

Luc De Raedt and Siegfried Nijssen

Department of Computer Science, Katholieke Universiteit Leuven, Belgium

Abstract. Today there is only little support for developing software that incorporates a machine learning or a data mining component. To alleviate this situation, we propose to develop programming languages for machine learning and data mining. We also argue that such languages should be declarative and should be based on constraint programming modeling principles. In this way, one could declaratively specify the problem of machine learning or data mining problem of interest in a high-level modeling language and then translate it into a constraint satisfaction or optimization problem, which could then be solved using particular solvers. These ideas are illustrated on problems of constraint-based itemset and pattern set mining.

1 Motivation

Today, machine learning and data mining are popular and mature subfields of artificial intelligence. The former is concerned with programs that improve their performance on specific tasks over time with experience and the later one with analyzing data in order to discover interesting patterns, regularities or models in the data. The two are intimately related in that machine learning often analyses data in order to learn, and that data mining often employs machine learning techniques to compute the regularities of interest, which explains why we shall not always distinguish these two fields. Significant progress in the past few years has resulted in a thorough understanding of different problem settings and paradigms and has contributed many algorithms, techniques, and systems that have enabled the development of numerous applications in science as well as industry.

Despite this progress, developing software that learns from experience or analyzes data remains extremely challenging because there is only little support for the programmer. Current support is limited to the availability of some software libraries [21] and the existence of data mining tools such as Weka and Orange, most of which only support the most common tasks of machine learning and data mining. Using these libraries for a specific application requires at best a thorough understanding of the underlying machine learning principles and algorithms; at worst it is impossible because the tool does not directly support the targeted learning or mining task. What is lacking is a direct support for the programmer of the machine learning software. This has motivated Tom Mitchell

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 25–32, 2011.

to formulate the following long term research question in his influential essay *The Discipline of Machine Learning* [15]:

> *Can we design programming languages containing machine learning primitives? Can a new generation of computer programming languages directly support writing programs that learn? ... Why not design a new computer programming language that supports writing programs in which some subroutines are hand-coded while others are specified as to be learned?*

This question is not really new; it represents an old but still unrealized dream that has been raised in a number different ways and contexts throughout the history of artificial intelligence. For instance, some *adaptive programming languages* [1,20] have been developed that embed hierarchical reinforcement learning modules in programming languages, while *probabilistic programming languages* aim at integrating graphical models and uncertainty reasoning into programming languages [9,4]. Other endeavors related to this question concern *inductive query languages*, which extend database query languages with the ability to declaratively query for patterns and models that in a database; these patterns and models become 'first class citizens' and the idea is to tightly integrate data mining inside databases; this has been the topic of a lot of research since the introduction of the vision by Iemielinski and Mannila [13]; cf. [3], and *automatic programming* [2], *inductive logic programming* [16,5] and *program synthesis by sketching* [12] which all attempt to synthesize in one way or another programs from examples of their input and output behavior.

While all these approaches have contributed important new insights and techniques, we are still far away from programming languages and primitives that support the writing and integration of programs for machine learning problems that arise in many applications.

In this extended abstract, we outline some principles and ideas that should allow us to alleviate this situation and we illustrate them using a particular example taken from our work on combining itemset mining and constraint programming [18,17]. Using this preliminary work, some interesting directions for further research are pointed out.

2 Machine Learning and Data Mining as Constraint Satisfaction and Optimization Problems

What we believe is necessary to realize Mitchell's vision, is a way to *declaratively* specify **what** the underlying machine learning problem is rather than outlining **how** that solution should be computed. Thus a number of modeling and inference primitives should be provided that allow the programmer to declaratively specify machine learning and data mining problems. This should be much easier than implementing the algorithms that are needed to compute solutions to these problems. Contemporary approaches to machine learning and data mining are too *procedural*, that is, they focus too much on the algorithms and the optimizations that are necessary to obtain high performance on specific tasks and

datasets. This makes it hard to identify common primitives and abstractions that are useful across a wide range of such algorithms. Yet abstraction is necessary to cope with the complexity of developing software. To make abstraction of the underlying algorithms we believe it is useful to assume that

Machine learning and data mining tasks can be declaratively expressed as constraint satisfaction and optimisation problems.

This assumption can be justified by observing that it is common practice to define machine learning tasks as those of finding an approximation $\hat{f}$ of an unknown target function f from data D such that

1. $\hat{f}$ belongs to a particular hypothesis space $\mathcal{H}$, that is, $\hat{f} \in \mathcal{H}$;
2. $\hat{f}$ is a good approximation of the target function f on the training data, that is, $\hat{f}(D) \approx f(D)$; and/or
3. $\hat{f}$ scores best with regard to a scoring function $score(f, D)$, that is, $\hat{f} = \arg\max_{f \in \mathcal{H}} score(f, D)$.

This type of problem is essentially a constraint satisfaction and optimization problem where the requirements $\hat{f} \in \mathcal{D}$ and $\hat{f}(D) \approx f(D)$ impose *constraints* on the possible hypotheses and the second requirement $\hat{f} = \arg\max_{f \in \mathcal{H}} score(f, D)$ involves the optimization step. In data mining, this is often formulated as computing a theory $Th(H, D, q) = \{f \in H | q(f, D) \text{ is true}\}$, where H is the space of possibly hypotheses, D the dataset and q specifies the constraints and optimization criteria [14].

We shall refer to the ensemble of constraints and optimization criterion as the *model* of the learning task. Models are almost by definition declarative and it is useful to distinguish *the constraint satisfaction* problem, which is concerned with finding a solution that satisfies all the constraints in the model, from the *optimization problem*, where one also must guarantee that the found solution be optimal w.r.t. the optimization function. Examples of typical constraint satisfaction problems in our context include local pattern mining, where the constraints impose for instance a minimum frequency threshold, and concept-learning, where the hypothesis should be consistent w.r.t. all examples. Typical optimization problems include the learning of support vector machines, where one wants to minimize the loss, and the parameters of a graphical model, where one wants to maximize the likelihood.

3 Declarative Programming for Machine Learning and Data Mining

Specifying a machine learning or data mining problem as a constraint satisfaction and optimization problem enables us to treat machine learning and data mining problems as any other constraint satisfaction and optimization problem. General methodologies for solving wide ranges of constraint satisfaction problems, as well

as the inclusion of these methodologies in programming languages, have been well studied within the field of constraint programming since the early 90s [19].

Applying the modeling principles of constraint programming to machine learning and data mining leads naturally to a layered approach in which one can distinguish:

the modeling (M) language is the most abstract language, which allows us to declaratively specify the problem; at this level the machine learning or data mining problem is encoded in a similar way as that used by machine learning and data mining researchers for specifying problems of interest;

the constraint satisfaction and optimization (CSO) language is a lower level language for specifying constraint satisfaction and optimization problems at an operational solver level, that is, at this level the problem is encoded in a way that is understood by solvers, and that may also include some procedural elements;

the programming (P) language is the (traditional) programming language which serves as the host language; at this level one can outline how to compute the inputs for the machine learning and data mining models and how to process the outputs.

In constraint programming, Essence [6] is an example of a modeling language. It allows one to specify combinatorial problems in almost the same way at that used in textbooks (such as Garey and Johnsson's [7]). The challenge is to translate these specifications into models at the CSO level, so that solutions can be computed by existing solvers. Finally, these solvers are embedded inside traditional programming languages.

A benefit of this approach is that it decouples the modeling language from the solver. In this regard, not only constraint programming solvers, but also satisfiability solvers or integer programming solvers could be used as a backend, where applicable.

To illustrate these ideas, consider the following example pptaken from our work on combining constraint programming and pattern mining [18,17], which fits within this paradigm. More specifically, we show the M and CSO-level specifications in Algorithms 1 and 2 for frequent itemset mining. This involves finding sets of items that frequently occur in a set of transactions. Frequent itemset mining is probably the simplest and best studied data mining problem.

Algorithm 1. Frequent Item-Set Mining at the M-level

```
1: given NrT, NrI : int                                   ▷ # transactions, # item
2: given D: matrix of boolean indexed by [int(1 ..NrT), int (1 .. NrI)]  ▷ the dataset
3: given Freq: int                                        ▷ frequency threshold
4: find  Items: matrix of boolean indexed by [int(1 ..NrI)]
5: such that  frequency(Items, D) ≥ Freq.
```

Algorithm 1 directly encodes the frequent itemset mining problem at the M-level. The CSO-level, illustrated in Algorithm 2, is a much more detailed

Algorithm 2. Frequent Item-Set Mining at the CSO-level

```
 1: given NrT, NrI : int                                          ▷ # transactions, # item
 2: given D:matrix of boolean indexed by [int(1 ..NrT), int (1 .. NrI)]   ▷ the dataset
 3: given Freq: int                                               ▷ frequency threshold
 4: find  Items: matrix of boolean indexed by [int(1 ..NrI)]
 5: find  Trans: matrix of boolean indexed by [int(1 ..NrT)]
 6: such that
 7:  for all t:int(1 .. NrT)                                      ▷ coverage constraint
 8:          Trans(t) <=>((sum i: int(1..NrI). !D[t,i]*Items[i]) <=0)
 9:  for all i:int(1 .. NrI)                                      ▷ frequency constraint
10:          Items(i) =>((sum t: int(1..NrT). D[t,i]*Trans[t]) >=Freq)
```

level model which provides an efficient encoding that can almost directly be written down in the primitives supported by constraint programming systems such as Gecode [8].[1] In the CSO formulation of the problem, one searches for a combination of two vectors Items and Trans such that 1) the transaction-set encoded by Trans corresponds exactly to all transactions in the dataset that are covered by the itemset encoded by Items (the coverage constraint); and 2) the itemset is frequent; cf. [18,17] for more details.

In a series of papers [18,17] we have shown that the same declarative constraint programming principles can not only be applied to frequent itemset mining but also to a wide variety of constraint-based pattern mining tasks such as finding maximal, closed, discriminative itemsets, ... This often involves only minor changes to the constraints. For instance, finding maximal frequent itemsets involves changing the "=>" implication in the frequency constraint in a "<=>" double implication, which shows the flexibility and power of the constraint programming approach. In addition, we have studied an extension of the pattern mining tasks to mining sets of k patterns [10] that satisfy constraints and we have shown that several well-known data mining tasks such as concept-learning, tiling, redescription mining and a form of conceptual clustering can be modeled within this framework.

Compared to using pattern mining algorithms present in libraries such as Weka, the benefit of the proposed approach is that it is easier to extend with further constraints and that it is easier to extend towards other settings: whereas in the procedural approach it is necessary to modify the algorithm in the library itself, in our proposed approach it suffices to change the high-level model of the problem, where any constraint provided in the modeling language can be used. For instance, by including statistical tests as a primitive in a modeling language, we could easily extend frequent itemset mining towards finding statistically relevant itemsets without implementing a new algorithm [17].

[1] The notation used in Algorithm 2 deviates from the actual one used by Gecode, as Gecode models problems in the C++ host language. It closely mirrors the way constraints CSO problems can be specified in Gecode, however.

4 Challenges

Whereas our previous work showed that constraint programming provides an interesting approach to addressing basic itemset mining problems, extending this approach towards other data mining and machine learning tasks faces several important challenges.

First, most current constraint programming systems are *finite domain* solvers, that is, they solve problems in which all variables are discrete and finite. In data mining and machine learning many tasks involve numerical computation and optimization, for instance, in statistical and Bayesian learning. Most current CP solvers are not well equipped to solve such numerical problems and do not yet provide a suitable language at the CSO-level. To deal with this issue in a scalable and usable manner, solvers are needed that also support numerical primitives such as convex optimization.

Fortunately, a key feature of constraint programming systems is their extendibility towards new constraints. A core area of CP research is that of developing propagators for new *global constraints*, i.e. constraints that involve many variables. An initial step could be to develop global constraints, as well as their propagators, for a number of showcase applications in mining and learning, such as clustering and Bayesian modeling; initially, these constraints could operate on discrete decision variables, while later on adding non-discrete decision variables can be considered.

Second, many data mining and machine learning problems are computationally hard so that it cannot be expected that a global solution can be calculated in reasonable time. In many cases, this might not even be needed and finding a locally optimal solution is sufficient. A solver is needed in which the specified problem is solved using local search or heuristic search. Ideally this solver would make reasonable choices with respect to how to perform the local search by itself, and only limited input by the user is needed at the P-level. Also here programming languages for local search under development in the constraint programming community could provide a useful starting point [11].

Third, even in the discrete case the current CSO modeling languages are not well adapted to machine learning and data mining tasks. In any language for mining and learning one would expect support for basic concepts such as datasets, coverage, and error, but there is currently no language at the M-level which supports these and makes modeling easy. Developing a suitable language at the M-level cannot be seen independently from developments at the other levels. The M-level language should be such that automatically mapping it to an appropriate model at the CSO-level is feasible; if new primitives are needed in underlying solvers, their implementation should be made easy. Also at the M-level there are significant challenges in designing effective and declarative high level modeling primitives; for instance, to model statistical and Bayesian learning approaches.

Fourth, a remaining challenge is that of dealing with structured data. When the data consists of networks, graphs, text, or logical descriptions, it is not clear how current solvers can be applied; it may be that new solvers are needed

(CSO-level), or that the mapping from the M-level to the CSO-level needs further study, exploiting results in grounding logical formulas, for instance.

5 Conclusions

In this extended abstract, it has been argued that programming languages for machine learning and data mining can be developed based on principles of constraint programming. This would involve declarative models specifying the machine learning or data mining problem at hand, and then, translating it into a lower level constraint satisfaction and optimization problem that can be solved using existing solvers. These preliminary ideas have been illustrated using our existing work on constraint programming and pattern mining. They leave open a large number of interesting research questions.

Acknowledgements

The authors would like to thank Tias Guns, Angelika Kimmig, and Guy Van den Broeck for interesting discussions about this work. Siegfried Nijssen is supported by the Research Foundation Flanders.

References

1. Andre, D., Russell, S.J.: Programmable reinforcement learning agents. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems, vol. 13, pp. 1019–1025 (2000)
2. Biermann, A., Guiho, G., Kodratoff, Y. (eds.): Automatic Program Construction Techniques. Macmillan, Basingstoke (1984)
3. Boulicaut, J.-F., De Raedt, L., Mannila, H. (eds.): Constraint-Based Mining and Inductive Databases. LNCS (LNAI), vol. 3848. Springer, Heidelberg (2006)
4. De Raedt, L., Frasconi, P., Kersting, K., Muggleton, S.H. (eds.): Probabilistic Inductive Logic Programming. LNCS (LNAI), vol. 4911, pp. 1–27. Springer, Heidelberg (2008)
5. De Raedt, L.: Logical and Relational Learning. Springer, Heidelberg (2008)
6. Frisch, A.M., Harvey, W., Jefferson, C., Hernández, B.M., Miguel, I.: Essence: A constraint language for specifying combinatorial problems. Constraints 13(3), 268–306 (2008)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, San Francisco (1979)
8. Gecode Team. Gecode: Generic constraint development environment (2006), http://www.gecode.org
9. Getoor, L., Taskar, B. (eds.): An Introduction to Statistical Relational Learning. MIT Press, Cambridge (2007)
10. Guns, T., Nijssen, S., De Raedt, L.: k-pattern set mining under constraints. Technical Report CW 596, Department of Computer Science, Katholieke Universiteit Leuven (2010)

11. Van Hentenreyck, P., Michel, L.: Constraint-based Local Search. The MIT Press, Cambridge (2005)
12. Hu, Z. (ed.): APLAS 2009. LNCS, vol. 5904. Springer, Heidelberg (2009)
13. Imielinski, T., Mannila, H.: A database perspectivce on knowledge discovery. Communications of the ACM 39(11), 58–64 (1996)
14. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)
15. Mitchell,T.: The discipline of machine learning.Technical Report CMU-ML-06-108, Carnegie Mellon University (2006)
16. Muggleton, S., De Raedt, L.: Inductive logic programming: Theory and methods. Journal of Logic Programming 19/20, 629–679 (1994)
17. Nijssen, S., Guns, T., De Raedt, L.: Correlated itemset mining in ROC space: a constraint programming approach. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 647–656 (2009)
18. De Raedt, L., Guns, T., Nijssen, S.: Constraint programming for itemset mining. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, pp. 204–212 (2008)
19. Rossi, F., van Beek, P., Walsh, T.: Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science Inc., Amsterdam (2006)
20. Simpkins, C., Bhat, S., Isbell Jr., C.L., Mateas, M.: Towards adaptive programming: integrating reinforcement learning into a programming language. In: Harris, G.E. (ed.) Proceedings of the 23rd Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, pp. 603–614 (2008)
21. Sonnenburg, S., Braun, M.L., Ong, C.S., Bengio, S., Bottou, L., Holmes, G., LeCun, Y., Müller, K.-R., Pereira, F., Rasmussen, C.A., Rätsch, G., Schölkopf, B., Smola, A., Vincent, P., Weston, J., Williamson, R.: The need for open source software in machine learning. Journal of Machine Learning Research 8, 2443–2466 (2007)

The Extraction Method of DNA Microarray Features Based on Modified *F* Statistics vs. Classifier Based on Rough Mereology

Piotr Artiemjew

Department of Mathematics and Computer Science
University of Warmia and Mazury
Olsztyn, Poland
artem@matman.uwm.edu.pl

Abstract. The paradigm of Granular Computing has emerged quite recently as an area of research on its own; in particular, it is pursued within the rough set theory initiated by Zdzisław Pawlak. Granules of knowledge can be used for the approximation of knowledge. Another natural application of granular structures is using them in the classification process. In this work we apply the granular classifier based on rough mereology, recently studied by Polkowski and Artiemjew 8_v1_w4 algorithm in exploration of DNA Microarrays. An indispensable element of the analysis of DNA microarray are the gene extraction methods, because of their high number of attributes and a relatively small number of objects, which in turn results in overfitting during the classification. In this paper we present one of our approaches to gene separation based on modified F statistics. The modification of F statistics, widely used in binary decision systems, consists in an extension to multiple decision classes and the application of a particular method to choose the best genes after their calculation for particular pairs of decision classes. The results of our research, obtained for modified F statistics, are comparable to, or even better than, the results obtained in other methods with data from the Advanced Track of the recent DNA Microarray data mining competition.

Keywords: rough mereology, granular computing, rough sets, DNA microarrays, features extraction.

1 Introduction

The last few years have seen a growing interest in the exploration of DNA microarrays; the more so due to some meaningful competitions, see [17]. A number of researchers have attempted to find effective gene extraction methods and classifiers in order to predict particular scientific problems. An exemplary application can be the ability to detect some illnesses, or predict vulnerability to some diseases, and distinguish some organisms' features or types. The main motivation to

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 33–42, 2011.

use our granular methods in DNA microarray exploration was our participation in the discovery challenge, see [17] at TunedIt platform. Our algorithm, based on the modified Fisher method [7] with our weighted voting classifier 8_v1_w4 see [14], reached eighteenth place on the basic track of this competition, and was worse only by 3.491 per cent balanced accuracy than the winner. Since that time we have been carrying out intensive research on new methods of gene extraction, and we have created more than 20 new methods of gene extraction. One of the effects of our work was the idea of DNA gene extraction methods based on modified F statistics.

This work is devoted to the classification problems of DNA arrays, with the use of the methods of granular computing presented in [12], [13]. In the Introduction we briefly show the idea of DNA microarrays, and define the basic concepts of granular computing in the sense of Polkowski , op.cit., and recall the idea of granular classification ([12], [13], [14]).

1.1 DNA Microarrays

The complementary DNA microarray is the most commonly used type of DNA microarrays, which is cheaper than other types of medical microarrays. The basic information about DNA microarray can be found in [4], [5], [6] and [15].

The DNA microarray is a tool which provides the means of measuring the gene expression on a mass scale, by simultaneously examining up to 40000 DNA strands with respect to their hybridization with complementary DNA (cDNA).

This analysis technique is widely applied in genome sequencing, for example the recognition of genes responsible for specific illnesses, etc. From the classification point of view, each gene can be regarded as an *attribute* and its value is the intensity of the bond with cDNA. A large number of attributes calls for new methods of data analysis, and in this paper we apply methods of granular classification, especially the method of weight incrementation in weighted voting by residual classifiers, as proposed in [2] and [14].

1.2 Basic Notions of Rough Set Theory and Granular Theory

In the light of rough set theory, knowledge can be represented by means of information or decision systems. An *information system* is defined as a pair (U, A) where U is a universe of *objects*, and A is a set of *attributes*; a *decision system* is defined as triple (U, A, d) where $d \notin A$ is a *decision*. Objects in u are represented by means of information sets: $Inf_A(u) = \{(a = a(u)) : a \in A\}$ is the *information set* of the object u; the formula $(a = a(u))$ is a particular case of a *descriptor* of the form $(a = v)$ where v is a value of the attribute $a \in A \cup \{d\}$.

Decision rules are expressions of the form

$$\bigwedge_{a \in A} (a = a(u)) \Rightarrow (d = d(u)) \tag{1}$$

In the classic meaning, the granulation of knowledge in information/decision systems consists of partitioning the set of objects U into classes of the *indiscernibility relation*

$$IND(A) = \{(u, v) : a(u) = a(v) \text{ for each } a \in A\} \tag{2}$$

Each class $[u]_A = \{v \in U : (u, v) \in IND(A)\}$ is interpreted as an elementary granule, and unions of elementary granules are granules of knowledge. Thus granulation, in this case, means forming aggregates of objects which are indiscernible over sets of attributes.

Rough inclusions, due to [11,12], are relations which in natural language can be expressed by saying that '*an object x is a part of an object y to a degree of r*'. A formal description of a rough inclusion is as a relation,

$$\mu \subseteq U \times u \times [0, 1] \tag{3}$$

In [12], [13], some methods for inducing rough inclusions in information/decision systems were introduced, from which we apply in this paper methods based on voting by test objects by means of weights computed with the help of residual rough inclusions, which we will now discuss.

Granular computing, introduced by Zadeh [18], consists in replacing objects with 'clumps of objects' collected together by means of a similarity relation, and in computing using these aggregates. In our setting, granules are formed by means of rough inclusions in the way pointed to in [12], see a survey in [13]. In formal terms, for a rough inclusion μ, an object u, and a real number $r \in [0, 1]$, a *granule about u of radius r*, $g(u, r)$ is defined as follows,

$$g(u, r) = \{v \in U : \mu(v, u, r)\} \tag{4}$$

1.3 Granular Classification of Knowledge

This type of granules may be applied in synthesis of a classifier in the way first proposed in [12]. The idea consists of fixing a radius of granulation r, and computing granules $g(u, r)$ for all objects $u \in U$. From the set of all granules a covering $C(U, r)$ is chosen, usually by means of a random choice. For each granule $g \in C(U, r)$, factored values $\overline{a}(g)$ of attributes a on g are computed, usually by means of majority voting, with random resolution of ties. The decision system $(U(C, r), \{\overline{a} : a \in A\}, \overline{d})$ is called a *granular resolution* of the initial decision system (U, A, d). For the granular resolution, various methods known in rough sets or elsewhere for classifier synthesis can be applied. The main features of this approach, see [2,3], [14], are: noise reduction - resulting in higher accuracy of classification - and classifier size reduction, resulting in much smaller number of classifying rules.

In the next section we describe in more detail rough inclusions used in this work along with their usage in analysis of DNA microarrays.

2 Application of Residual Rough Inclusions

For the decision system (U, A, d), we outline a rough inclusion based on the notion of a residuum of a t–norm.

2.1 Residua of T–Norms, Residual Rough Inclusions

The function $T : [0,1] \times [0,1] \rightarrow [0,1]$ which is symmetric, associative, increasing in each coordinate, and subject to boundary conditions: $T(x,0) = 0, T(x,1) = x$, is a t–norm, see, e.g, [8]. Examples of t–norms are,

1. (the Łukasiewicz t–norm) $L(x,y) = max\{0, x+y-1\}$.
2. (the Product t–norm) $P(x,y) = x \cdot y$.
3. (the Minimum t–norm) $M(x,y) = min\{x,y\}$.

By a *residuum* $x \Rightarrow_T y$ of a t–norm T, a function is meant, defined by means of,

$$x \Rightarrow_T y \geq r \ if \ and \ only \ if \ T(x,r) \leq y \tag{5}$$

As all t–norms L, P, M are continuous, in their cases, the residual implication is given by the formula,

$$x \Rightarrow_T y = max\{r : T(x,r) \leq y\} \tag{6}$$

Residual rough inclusions on the interval $[0,1]$ are defined, see, eg, [13] as,

$$\mu_T(x,y,r) \ if \ and \ only \ if \ x \Rightarrow_T y \geq r \tag{7}$$

2.2 A Voting Scheme for Decision Value Assignment

In classifier synthesis, this rough inclusion, e.g., induced by the Łukasiewicz t–norm L, is applied in the following way. As usual, the data set is split into *training set* and the *test set*. For a test object u, training objects v vote for decision value at u by means of weights

$$w(v,u,\varepsilon) = dis_\varepsilon(u,v) \Rightarrow_T ind_\varepsilon(u,v) \tag{8}$$

For each decision value v_d, a parameter,

$$Param(v_d) = \sum_{\{v \in U_{trn} : d(v) = v_d\}} w(v,u,\varepsilon) \tag{9}$$

is computed and the decision value assigned to u is $v_d(u)$ with the property that

$$v_d(u) = min\{Param(v_d) : v_d\} \tag{10}$$

We have introduced basic facts about our approach, and now we return to our analysis of DNA microarrays.

3 DNA Microarray Features Extraction Method

The main purpose of this work is to present a selected gene extraction method based on modified F statistics - by using a classifier based on mereological granules. The data presented in this paper can be interpreted without context, because all the decision classes of the examined DNA microarrays are classified in a general sense, as one big decision system.

The huge amount of information obtained from DNA microarrays, due to the large number of gene–attributes, needs some preparatory methods in order to reduce this amount of information. We attempt to choose the genes that best separate the decision classes. Our approach to the separation of classes in this paper is as follows.

We have applied here F statistics, extended over multiple decision classes, which are well-known for the separation of the two decision classes.

Features Extraction Method Based on Modified F Statistics Method: Case6 (MSF6). For the decision system (U, A, d), where $U = \{u_1, u_2, ..., u_n\}$, $A = \{a_1, a_2, ..., a_m\}$, $d \notin A$, classes of d: $c_1, c_2, ..., c_k$, we propose to obtain the rate of separation of the gene $a \in A$ for pairs of decision classes c_i, c_j, where $i, j = 1, 2, ..., k$ and $i \neq j$ in the following way. We let,

$$F_{c_i,c_j}(a) = \frac{MSTR_{c_i,c_j}(a)}{MSE_{c_i,c_j}(a)} \tag{11}$$

$$C_i^a = \{a(u) : u \in U \text{ and } d(u) = c_i\}, C_j^a = \{a(v) : v \in U \text{ and } d(v) = c_j\}.$$

$$\overline{C}_i^a = \frac{\{\sum a(u) : u \in U \text{ and } d(u) = c_i\}}{card\{C_i^a\}}, \overline{C}_j^a = \frac{\{\sum a(v) : v \in U \text{ and } d(v) = c_j\}}{card\{C_j^a\}}$$

$$\overline{C}_{i,j}^a = \frac{\{\sum a(u) : u \in U \text{ and } (d(u) = c_i \text{ or } d(u) = c_j)\}}{card\{C_i^a\} + card\{C_j^a\}},$$

$$MSTR_{c_i,c_j}(a) = card\{C_i^a\} * (\bar{C}_i^a - \bar{C}_{i,j}^a)^2 + card\{C_j^a\} * (\bar{C}_j^a - \bar{C}_{i,j}^a)^2$$

$$MSE_{c_i,c_j}(a) = \frac{\sum_{l=1}^{card\{C_i^a\}}(a(u_l) - \bar{C}_i^a) + \sum_{m=1}^{card\{C_j^a\}}(a(v_m) - \bar{C}_j^a)}{card\{C_i^a\} + card\{C_j^a\} - 2}$$

$$\text{where } u_l \in C_i^a,\ l = 1, 2, ..., card\{C_i^a\},\ v_m \in C_j^a, m = 1, 2, ..., card\{C_j^a\}$$

After the rate of the separation $F_{c_i,c_j}(a)$, are computed for all genes $a \in A$ and all pairs of decision classes c_i, c_j, where $i \neq j$ and $i < j$ genes are sorted in decreasing order of , $F_{c_i,c_j}(a)$

$F^1_{c_{i_1},c_{i_2}} > F^2_{c_{i_1},c_{i_2}} > ... > F^{card\{A\}}_{c_{i_1},c_{i_2}}$, where $i_1 \in \{1,2,...,k-1\}$ and $i_2 \in \{i_1+1,...,k\}$

Finally, for experiments we have chosen the fixed number of genes from the sorted list by means of the procedure,

Procedure
Input data
$A' \leftarrow \emptyset$
$iter \leftarrow 0$
for $i = 1,2,...,card\{A\}$ **do**
  for $j_1 = 1,2,...,k-1$ **do**
    for $j_2 = j_1+1,...,k$ **do**
      if $F_{c_{j_1},c_{j_2}}(a) = F^i_{c_{j_1},c_{j_2}}(a)$ and $a \notin A'$ **then**
        $A' \leftarrow a$
        $iter \leftarrow iter + 1$
        if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
          BREAK
        end if
      end if
    end for
    if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
      BREAK
    end if
  end for
  if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
    BREAK
  end if
end for
return A'

4 Augmented Weighted Voting by Granules of Training Objects

The voting scheme proposed in sect. 2.2 is here augmented along the lines of [2]. The idea is to increase or decrease weights depending on the case, as shown in five variants (as Algorithms 8_v1.1, v1.2, v1.3, v1.4, v1.5 of [2]). These variants are described in [1], [14], but in this work we use only the best algorithm among those studied, variant 8_v1.4.

The procedure of chosen algorithm is as follows:

Step 1. The training decision system (U_{trn}, A, d) and the test decision system (U_{tst}, A, d) have been input, where U_{tst}, U_{trn} are, respectively, the test set and the training set, A is a set of attributes, and d is a decision.

Step 2. max_attr_a and min_attr_a have been found from the training data set, where max_attr_a, min_attr_a are, respectively, the maximal and the minimal value of attribute a on the training set.

Step 3. A chosen value of ε (determining attribute similarity degree) has been input.

Step 4. Classification of testing objects by means of weighted granules of training objects is done as follows:

For all conditional attributes $a \in A$, training objects $v_p \in U_{trn}$, where $p \in \{1, ..., card\{U_{trn}\}$ and test objects $u_q \in U_{tst}$, where $q \in \{1, ..., card\{U_{tst}\}$, for $train_a = max\_attr_a - min\_attr_a$ and $||a(u_q) - a(v_p)|| = \frac{|a(u_q)-a(v_p)|}{train_a}$ we compute

Subcase a) *If* $||a(u_q) - a(v_p)|| \geq \varepsilon$, *then*

$$w(u_q, v_p) = w(u_q, v_p) + \frac{|a(u_q) - a(v_p)|}{train_a * (\varepsilon + ||a(u_q) - a(v_p)||)}$$

i. e.,

$$w(u_q, v_p) = w(u_q, v_p) + \frac{|a(u_q) - a(v_p)|}{train_a * \varepsilon + |a(u_q) - a(v_p)|}$$

Subcase b) *If* $||a(u_q) - a(v_p)|| < \varepsilon$, *then*

$$w(u_q, v_p) = w(u_q, v_p) + \frac{|a(u_q) - a(v_p)|}{train_a * \varepsilon}$$

After weights in either Case are computed - for a given test object u_q and each training objects v_p - the voting procedure comprises computing values of parameters,

$$Param(c) = \sum_{\{v_p \in U_{trn} : d(v_p) = c\}} w(u_q, v_p), \tag{12}$$

for $\forall c$, decision classes.

Finally, the test object u_q is classified to the class c^* with a minimal value of Param(c).

After all test objects u_q are classified, quality parameters Total accuracy and Total coverage are computed.

The results for our algorithms with real DNA microarrays (see Table 1 from Advanced Track of Discovery Challenge see [16] and [17]) are reported in the next section.

5 The Results of Experiments with Leave One Out Method for Sample DNA Microarray Data

As we have studied, DNA microarrays contain unequal and small significant decision classes - see Table 1 - which is why we are evaluating results by a balanced accuracy parameter,

$$B.acc = \frac{acc_{c_1} + acc_{c_2} + ... + acc_{c_k}}{k} \tag{13}$$

Due to considerations of space, only an exemplary test can be discussed here. We apply our best classification algorithm 8_v1.4 among those studied [1] based on weighed voting with fixed parameter $\varepsilon = 0.01$, and our feature extraction method with Leave One Out method (LOO). For Leave One Out method a confusion matrix is built, in which the tested objects from all folds are treated as one test decision system. The motivation to use the Leave One Out method can be found, among other places in [9] and [17]. These papers prove the effectiveness and almost unbiased character of this method. Another argument proving its effectiveness it that FS+LOO model was successfully used for microarray data by the winners of the Advanced Track competition [17].

Table 1. An information table of the examined data sets (see [16]); data1 = anthracyclineTaxaneChemotherapy, data2 = BurkittLymphoma, data3 = HepatitisC, data4 = mouseType, data5 = ovarianTumour, data6 = variousCancers_final

Data.name	*No.of.attr*	*No.of.obj*	*No.of.dec.class*	*The.dec.class.details*
data1	61359	159	2	1(59.7%), 2(40.2%)
data2	22283	220	3	3(58.1%), 2(20%), 1(21.8%)
data3	22277	123	4	2(13.8%), 4(15.4%), 1(33.3%), 3(37.3%)
data4	45101	214	7	3(9.8%), 2(32.2%), 7(7.4%), 6(18.2%), 5(16.3%), 4(9.8%), 1(6%)
data5	54621	283	3	3(86.5%), 1(6.3%), 2(7%)
data6	54675	383	9	3(6.2%), 2(40.4%), 4(10.1%), 7(5.2%), 5(12.2%), 6(10.9%), 8(4.1%), 9(4.6%), 10(5.7%)

5.1 The Results for Our Gene Extraction Method

DNA microarray gene separation method MSF6 based on modified statistic F produces one of the best average results in a global sense from among all the methods that we have studied. On the basis of average results for our best method - see Table 2 - we can conclude that the best balanced accuracy 0.789 for all examined data has been obtained with only 50 genes. Table 4 presents the comparison of our best results and the results of the winners of Advanced Track discovery challenge - see [17]. It is evident that our methods are comparable to, or even better than, other methods. Balanced accuracy computed in all 28 decision classes of examined data is about 3 percent better than the best from Advanced Track [17].

Table 2. Leave One Out; The average balanced accuracy of classification for MSF6 algorithm; Examined data sets: all from Table 1; No.of.genes = number of classified genes, method = method's name

method\No.of.genes	10	20	50	100	200	500	1000
MSF6	**0.718**	**0.759**	**0.789**	**0.782**	**0.781**	**0.777**	**0.783**

Table 3. Leave One Out; 50 genes; The balanced accuracy of classification for all 28 decision classes with MSF6 algorithm; Examined data sets: all from Table 1; acc_b = Balanced Accuracy

data.class	acc_b	*data.class*	acc_b	*data.class*	acc_b	*data.class*	acc_b
*data*1.1	0.568	*data*3.1	0.927	*data*4.4	0.81	*data*6.4	0.538
*data*1.2	0.703	*data*3.3	0.87	*data*4.1	1	*data*6.7	1
*data*2.3	0.969	*data*4.3	0.952	*data*5.3	0.963	*data*6.5	0.809
*data*2.2	0.977	*data*4.2	0.536	*data*5.1	1	*data*6.6	0.714
*data*2.1	0.688	*data*4.7	0.438	*data*5.2	0.4	*data*6.8	0.938
*data*3.2	0.941	*data*4.6	0.359	*data*6.3	0.958	*data*6.9	0.833
*data*3.4	1	*data*4.5	0.629	*data*6.2	0.665	*data*6.10	0.909

Table 4. Average balanced accuracy; Modified F statistics vs Advanced Track results of the Discovery Challenge [17]; Examined data sets: all from Table 1; in case * Leave One Out result for 50 genes

method	*Balanced Accuracy*
*MSF6**	**0.789**
RoughBoy [17]	0.75661
ChenZe [17]	0.75180
wulala [17]	0.75168

6 Conclusions

The research results for our 8_v1_w4 classification method [2], [14] (with gene extraction MSF6 algorithm with examined data) are comparable to the best results from the Advanced Track of data mining contest see [17]. Those results have been evaluated by means of average balanced accuracy computed in all 28 decision classes of examined data. What follows from our experiments is that the essential element of gene separation methods is the way to choose the best genes after their calculation. In the case of the MSF6 method we choose genes which best separate particular pairs of decision classes one by one from all combinations, without the repetition of length 2 of decision classes.

The search is in progress for a theoretical explanation of the effectiveness of gene separation methods, based on F statistics, as well as work aimed at developing the theoretical description of these statistics, and will be reported.

Acknowledgements

The author wishes to express his thanks to Professor Lech Polkowski for his invaluable support and advice.

The research has been supported by a grant 1309-802 from the Ministry of Science and Higher Education of the Republic of Poland.

References

1. Artiemjew, P.: Classifiers based on rough mereology in analysis of DNA microarray data. In: Proceedings 2010 IEEE International Conference on Soft Computing and Pattern Recognition SocPar 2010. IEEE Press, Sergy Pontoise France (2010)
2. Artiemjew, P.: On strategies of knowledge granulation and applications to decision systems, PhD Dissertation, Polish Japanese institute of Information Technology, L. Polkowski, Supervisor, Warsaw (2009)
3. Artiemjew, P.: On Classification of Data by Means of Rough Mereological Granules of Objects and Rules. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 221–228. Springer, Heidelberg (2008)
4. Brown, M., Grundy, W., et al.: Knowledge-based analysis of microarray gene expression data by using support vector machines. University of California, Berkeley (1999)
5. Eisen, M.B., Brown, P.O.: DNA arrays for analysis of gene expression. Methods Enzymol 303, 179–205 (1999)
6. Furey, T.S., Cristianini, D.N., Bernarski, S.M., Haussler, D.: Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data. Bioinformatics 16, 906–914 (2000)
7. Gorecki, P., Artiemjew, P.: DNA microarray classification by means of weighted voting based on rough set classifier. In: Proceedings 2010 IEEE International Conference on Soft Computing and Pattern Recognition SocPar 2010, pp. 269–272. IEEE Computer Society, Sergy Pontoise (2010)
8. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
9. Molinaro, A.M., Simon, R., Pfeiffer, R.M.: Prediction error estimation: a comparison of resampling methods. Bioinformatics 21(15), 3301–3307 (2005)
10. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
11. Polkowski, L.: Toward rough set foundations. Mereological approach (a plenary lecture). In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 8–25. Springer, Heidelberg (2004)
12. Polkowski, L.: Formal granular calculi based on rough inclusions (a feature talk). In: Proceedings 2005 IEEE Int. Confrence on Granular Computing GrC 2005, pp. 57–62. IEEE Press, Los Alamitos (2005)
13. Polkowski, L.: A Unified Approach to Granulation of Knowledge and Granular Computing Based on Rough Mereology: A Survey. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 375–401. John Wiley & Sons, New York (2008)
14. Polkowski, L., Artiemjew, P.: On classifying mappings induced by granular structures. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 264–286. Springer, Heidelberg (2008)
15. Schena, M.: Microarray analysis. Wiley, Hoboken (2003)
16. http://tunedit.org/repo/RSCTC/2010/A
17. Wojnarski, M., Janusz, A., Nguyen, H.S., Bazan, J., Luo, C., Chen, Z., Hu, F., Wang, G., Guan, L., Luo, H., Gao, J., Shen, Y., Nikulin, V., Huang, T.-H., McLachlan, G.J., Bošnjak, M., Gamberger, D.: RSCTC'2010 Discovery Challenge: Mining DNA Microarray Data for Medical Diagnosis and Treatment. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 4–19. Springer, Heidelberg (2010)
18. Zadeh, L.A.: Fuzzy sets and information granularity. In: Gupta, M., Ragade, R., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 3–18. North Holland, Amsterdam (1979)

Attribute Dynamics in Rough Sets

Davide Ciucci

Dipartimento di Informatica, Sistemistica e Comunicazione
Università di Milano – Bicocca
Viale Sarca 336 – U14, I–20126 Milano Italia
ciucci@disco.unimib.it

Abstract. Dynamics with respect to attributes in an information (decision) table is studied. That is, we show how the main rough set instruments change when new knowledge (in form of new attributes) is available. In particular, we analyze the approximations, positive regions, generalized decision, reducts and rules. With respect to classification, an algorithm to update reducts and rules is presented and a preliminary study on rule performance is given.

1 Introduction

In [2] we gave a classification of dynamics in rough sets. The main distinction was between synchronous and asynchronous dynamics. In this second case, we characterized different ways to have an increase, or dually decrease of information, in time and with respect to three different approaches to rough sets: Information Tables, Approximation Spaces and Coverings. When considering Information Tables, we can have an increase of information with respect to objects, attributes and values (a formal definition will be given in the next section). Increase of information with respect to objects has been studied in [11], where an algorithm to incrementally update the reducts has been proposed. In [12] objects observed at different times are collected in a unique *temporal information system*, which is then studied, for instance with regards to changes of functional dependencies between attributes. In [3] increase of information with respect to values is addressed. The authors show how approximations and positive regions change when an unknown value becomes known and an algorithm to update reducts is also proposed. Grzymala-Busse [4] analyses in quantitative terms what happens when known values are turned to unknown. The result is that imprecise tables (i.e., with more unknown values) generate better rules.

Here we are focusing on attributes and study what happens when we add (or delete) an attribute to an Information Table. At first, we give some theoretical results. In particular, we study the behaviour of set approximations, positive regions and reducts. Then, we deal with rule performance in applications.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 43–51, 2011.

2 Preliminary Notions

2.1 Rough Sets Basis

Information Tables (or Information Systems) [7,9] are basic terms of rough set theory. They have been defined to represent knowledge about objects in terms of observables (attributes).

Definition 2.1. *An* Information Table *is a structure* $\mathcal{K}(X) = \langle X, A, val, F\rangle$ *where:*

- *the universe* X *is a non empty set of* objects*;*
- A *is a non empty set of* condition attributes*;*
- *val is the set of all* possible values *that can be observed for all attributes;*
- F *(called the* information map*) is a mapping* $F : X \times A \to val \cup \{*\}$ *which associates to any pair object–attribute, the value* $F(x, a) \in val$ *assumed by the attribute* a *for the object* x*. If* $F(x, a) = *$ *it means that this particular value is unknown.*

Let us note that we do not deal with different semantics of incomplete information tables, but simply take into account that for some reason a value can be missing, i.e., $F(x, a) = *$.
A *decision table* is an Information Table where the attributes are divided in two groups $A \cup D$, with A condition attributes and D decision attributes which represent a set of decisions to be taken given the conditions represented by A. Usually, $|D| = 1$.

Given an information (or decision) table, the indiscernibility relation with respect to a set of attributes $B \subseteq A$ is defined as

$$x\mathcal{I}_B y \quad \text{iff} \quad \forall a \in B,\ F(x, a) = F(y, a)$$

This relation is an equivalence one, which partitions X into equivalence classes $[x]_B$, our *granules* of information. Due to a lack of knowledge we are not able to distinguish objects inside the granules, thus not all subsets of X can be precisely characterized in terms of the available attributes B. However, any set $H \subseteq X$ can be approximated by a lower and an upper approximation, respectively defined as:

$$L_B(H) = \{x : [x]_B \subseteq H\} \tag{2.1a}$$
$$U_B(H) = \{x : [x]_B \cap H \neq \emptyset\} \tag{2.1b}$$

Other forms of imprecision arise in decision tables, since it may happen that two objects with the same conditions have different decision. In this case the decision table is said *non-deterministic*, and we can define the *generalized decision*:

$$\delta_B(x) = \{i : F(y, d) = i \text{ and } x\mathcal{I}_B y\}$$

Thus, in a non-deterministic situation, only a subset of objects can be precisely classified: the *positive region* of the decision table, defined as

$$POS_B(\mathcal{K}(X), d) = \cup L_B([x]_{\{d\}})$$

A decision table can be simplified by searching for a reduct: the smallest set of attributes which preserves the classification. More precisely, given a decision table, a set of attributes $B_1 \subseteq A$ is a *reduct* of $B_2 \subseteq A$ if

1. B_1 and B_2 generate the same generalized decision: for all objects $x \in X$, $\delta_{B_1}(x) = \delta_{B_2}(x)$;
2. A minimality condition holds: for all $C \subseteq B_2$ such that $\delta_C = \delta_{B_2}$, then $B_1 \subseteq C$.

Clearly, there can be more than one reduct for a given set B_2.

Finally, classification rules can be deduced by a reduct or directly computed by a proper algorithm, for instance LEM (and its descendants). Here, we are not going into details on how we can obtain rules, for an overview the reader is referred to [8]. We just denote a rule as $r : a_1 = v_1, \ldots a_n = v_n \rightarrow d_1$ or $d_2 \ldots$ or d_m, with the meaning that when conditions $a_i = v_i$ are satisfied than an object can belong to one of the decisions d_i. Of course, in the deterministic case we have $m = 1$, that is only one decision is possible.

2.2 Temporal Dynamics in Information Tables

If we consider an information system evolving in time, it may change in terms of objects, attributes, values or information map. We can describe three different situations where the knowledge increases going from time t to time $t + 1$ and they are formalized in the following way.

Definition 2.2. [2] *Let* $\mathcal{K}^{(t_1)}(X_1) = \langle X_1, A_1, val_1, F_1\rangle$ *and* $\mathcal{K}^{(t_2)}(X_2) = \langle X_2, A_2, val_2, F_2\rangle$*, with* $t_1, t_2 \in \mathbb{R}$*,* $t_1 \leq t_2$ *be two Information Tables. We will say that there is a* monotonic increase of information *from time* t_1 *to time* t_2

- *wrt* values *iff* $\mathcal{K}^{(t_1)}$ *and* $\mathcal{K}^{(t_2)}$ *are defined on the same set of objects, attributes and values and* $F_1(x, a) \neq *$ *implies* $F_2(x, a) = F_1(x, a)$.

- *wrt* attributes *iff* $X_1 = X_2$*, i.e.,* $\mathcal{K}^{(t_1)}$ *and* $\mathcal{K}^{(t_2)}$ *are defined on the same set of objects and* $A_1 \subseteq A_2$*,* $val_1 \subseteq val_2$ *and* $\forall a \in A_1, \forall x \in X_1, F_2(x, a) = F_1(x, a)$.

- *wrt* objects *iff* $\mathcal{K}^{(t_1)}$ *and* $\mathcal{K}^{(t_2)}$ *have the same set of attributes and values,* $X_1 \subseteq X_2$ *and* $\forall x \in X_1$*,* $F_2(x, a) = F_1(x, a)$.

In all the three cases we can also define a *decrease of knowledge* when the reverse orderings hold. In the sequel we focus on the increase/decrease with respect to attributes.

Example 2.1. In Table 1 we can see a monotone increase wrt attributes from time t_1 to time t_2 since the new attribute *Rooms* is added while the others do not change.

Table 1. Flats incomplete information systems

Observer at time t_1			
Flat	Price	Down-Town	Furniture
f_1	high	yes	*
f_2	high	yes	no
f_3	*	yes	no
f_4	*	*	yes

Observer at time t_2				
Flat	Price	Rooms	Down-Town	Furniture
f_1	high	2	yes	*
f_2	high	*	yes	no
f_3	*	2	yes	no
f_4	*	1	*	yes

3 Theoretical Results

Now, we give some results showing that, as expected, to a new attribute corresponds a deeper knowledge on the problem. At first, in proposition 3.1 (already given in [1] without proof), we see that an increase of knowledge with respect to attributes leads to better approximations.

Proposition 3.1. *Let $\mathcal{K}^{(t_0)}(X)$ and $\mathcal{K}^{(t_1)}(X)$ be two information systems characterized by a monotone increase of knowledge with respect to attributes from time t_0 to time t_1. Then*

$$L^{t_0}_{\mathcal{R}}(H) \subseteq L^{t_1}_{\mathcal{R}}(H) \subseteq H \subseteq U^{t_1}_{\mathcal{R}}(H) \subseteq U^{t_0}_{\mathcal{R}}(H)\,.$$

Sketch of Proof. With a new attribute, equivalence classes become smaller. Hence, from equations (2.1) more objects can belong to the lower approximation and less to the upper.

Let us note that the above proposition can be applied to more general rough-set models, for instance tolerance rough sets. Indeed, in order to apply it, it is sufficient to have a situation in which a new attribute induces a smaller granulation or not greater one.

As a consequence of proposition 3.1, also the number of objects that can be correctly classified (that is the positive region of the decision table) increases.

Proposition 3.2. *Let $\mathcal{K}^{(t_0)}(X)$ and $\mathcal{K}^{(t_1)}(X)$ be two information systems characterized by a monotone increase of knowledge with respect to attributes from time t_0 to time t_1. Then,*

$$POS_B(\mathcal{K}^{(t_0)}(X), d) \subseteq POS_B(\mathcal{K}^{(t_1)}(X), d)$$

Proof. It follows from definition and proposition 3.1.

As expected, generalized decisions have the opposite behaviour: for a given object, its generalized decision becomes smaller.

Proposition 3.3. *Let $\mathcal{K}^{(t_0)}(X)$ and $\mathcal{K}^{(t_1)}(X)$ be two information systems characterized by a monotone increase of knowledge with respect to attributes from time t_0 to time t_1. Then, for all $x \in X$:*

$$\delta_{A_{t_1}}(x) \subseteq \delta_{A_{t_0}}(x).$$

Proof. Also this result is due to a finer granulation of the objects when having more attributes.

3.1 Reducts Update

Now, let us consider the reducts. Supposing that $A_{t_0} \subseteq A_{t_1}$, it can happen that Red_{t_0}, the reduct of A_{t_0}, is a reduct of A_{t_1}. In this case the reduct at time t_1 does not change. On the contrary, if Red_{t_0} is not a reduct of A_{t_1}, we have that $Red_{A_{t_0}} \subseteq Red_{A_{t_1}}$ and the rules obtained at time t_1 are more precise, in the sense that they contain more conditions. If we desire to compute the reducts at time t_1 we can start from an existing one and update it, instead of re-calculating them all. For the sake of simplicity, let us suppose that we add only one attribute a from time t_0 to time t_1. A way to update a reduct is sketched in algorithm 1.

Algorithm 1. Update reducts

Require: A set of objects $X = \{x_1, x_2, \ldots x_n\}$, a reduct Red_{t_0}, a new attribute a
Ensure: An updated reduct

1: $i = 1$
2: **while** $\delta_{A_{t_1}}(x_i) = \delta_{A_{t_0}}(x_i)$ **do**
3: increase i
4: **end while**
5: **if** $i \neq n + 1$ **then**
6: add a to the reduct: $Red_{t_1} = Red_{t_0} \cup \{a\}$
7: **end if**

That is, we add a to the reduct Red_{t_0} if it enables to discern objects belonging to different decision classes which were equivalent with respect to attributes in Red_{t_0}. Clearly, if we add more than one attribute we can proceed by applying the above algorithm for each attribute. In this case, the order of considering the attributes will influence the result (heuristics can be found in literature [8]) and we may also loose some reduct respect to computing them from scratch. Another approach could be to ask if there exists a reduct at time t_1 which contains the reduct at time t_0. This can be solved in polynomial time similarly to the covering problem discussed in [6].

Example 3.1. Let us consider a simple example of a medical decision table (tables 2, 3), in which patients are characterized according to some attribute and a decision on their disease has to be taken.

At time t_0 a possible reduct is made of only one attribute: Red_{t_0} ={Pressure}. At time t_1 we add the attribute Temperature and we recompute the generalized decision. We have that $\delta_{A_{t_1}}(x_1) = \{A\} \neq \{A, NO\} = \delta_{A_{t_0}}(x_1)$. Hence, we add the attribute Temperature and obtain Red_{t_1} ={Pressure, Temperature}.

Table 2. Medical decision table, time t_0

Patient	Pressure	Headache	Muscle Pain	Disease
p_1	2	yes	yes	A
p_2	3	no	yes	B
p_3	1	yes	no	NO
p_4	2	no	yes	NO

Table 3. Medical decision table, time t_1

Patient	Pressure	Headache	Muscle Pain	Temperature	Disease
p_1	2	yes	yes	very high	A
p_2	3	no	yes	high	B
p_3	1	yes	no	normal	NO
p_4	2	yes	yes	high	NO

3.2 Rules Update

A similar method to the one described for reducts can be applied directly to rules. The deterministic rules are not changed, since the new attribute will not affect the classification. Imprecise rules can be improved if the new attribute is able to discern similar objects with different decisions. In algorithm 2 a rule is substituted by a set of new rules.

We add a new rule if (condition at line 4) the set of objects satisfying the conditions of the actual rule is not contained in one of the equivalence classes with respect to the new attribute. Let us note that the new rules are better than the hold ones in the sense that they have less elements in the right part.

Example 3.2. Let us consider the same decision tables of example 3.1. A rule at time t_0 is

$$r_0 : \text{Pressure} = 2, \text{Headache} = \text{yes}, \text{Muscle Pain} = \text{yes} \rightarrow \text{A or NO}.$$

The set of patients satisfying r_0 is $H = \{p_1, p_4\}$ and we get $[p_1]_{Temp} = \{p_1\}$, $[p_4]_{Temp} = \{p_4\}$. Since $H \not\subseteq [p_1]_{Temp}$ and $H \not\subseteq [p_4]_{Temp}$, at time t_1, we have two new rules:

r_1' : Pressure $= 2$, Headache $= yes$, Muscle Pain $= yes$, Temp. $= very\_high \rightarrow$ A

r_1'' : Pressure $= 2$, Headache $= yes$, Muscle Pain $= yes$, Temp. $= high \rightarrow$ NO.

We note that the attributes Headache and Muscle Pain are not decisive in this rules and they can be removed without affecting the result.

Algorithm 2. Update rules

Require: a decision table, a new attribute a, an imprecise rule of the form $r : a_1 = v_1, a_2 = v_2, \ldots a_n = v_n \rightarrow (d_1$ or d_2 or $\ldots$ or $d_m)$

Ensure: updated versions of the rule r

1: Compute $H =$ all the objects which satisfies the conditions of r
2: Partition H with respect to the attribute a in classes $[x]_a$
3: **for** all classes $[x]_a$ such that $[x]_a \cap H \neq \emptyset$ **do**
4: **if** $H \not\subseteq [x]_a$ **then**
5: generate the rule $a_1 = v_1, a_2 = v_2, \ldots a_n = v_n, a = F(x, a) \rightarrow \delta(x)$
6: **end if**
7:
8: **end for**

4 Experiments

From a theoretical standpoint, we expect that the rules computed with more attributes are more accurate. We can however wonder if, as in the case of values increase of knowledge [4], the less accurate rules have better performances. In order to verify this, we made some preliminary test. We used ROSE2[1] as our software framework. Rules are computed directly from the table without discretization which is performed directly inside the execution of MLEM algorithm [5]. All the datasets are taken from the UCI repository[2] and accuracy measures are obtained by a 10-fold cross validation procedure

The first data set taken into account is the well-known Iris one. In this case performances are measured with respect to the full set of attributes, then deleting one attribute (25% of the total) and two attributes (50%). The results are reported in table 4.

As can be seen in the last three rows, on average accuracy decreases as attributes diminish. That is, to a monotone decrease of knowledge with respect to attributes corresponds a decrease of accuracy. From this table we can also get some (indirect) indication on the dependence of the decision from the conditions. Indeed, deleting attributes 1 and 2 has a weaker impact than deleting 3 and 4, either separately or together. This means that the classification depend in a stronger manner on attributes 3 and 4, than on attributes 1 and 2.

[1] http://idss.cs.put.poznan.pl/site/rose.html
[2] http://archive.ics.uci.edu/ml/

Table 4. Iris dataset accuracy

Missing attributes	Accuracy
1,2	94.0
1,3	92.0
1,4	92.0
2,3	92.0
2,4	92.0
3,4	72.0
1	94.7
2	94.0
3	92.0
4	92.0
None	95.3
Average 25%	93.2
Average 50%	89.0

Table 5. Pima diabetes (on the left) and breast cancer (right) datasets accuracy

Missing attributes	Accuracy
1,2,3,4	70.1
2,3,4,5	67.6
3,4,5,6	72.4
4,5,6,7	69.9
5,6,7,8	69.1
2,6	65,62
5,7	72.8
1,4	74.4
3,8	70.6
2,4	68.5
None	73.4
Average 25%	70.4
Average 50%	69.8

Missing attributes	Accuracy
1,3,4,8	56.9
2,5,6,7	60.0
1,3,5,6	69.2
2,4,7,8	59.2
1,2,3,7	73.7
1,4	69.1
7,8	65.9
3,5	66.6
2,6	68.5
1,6	64.9
None	66.7
Average 25%	67.0
Average 50%	63.8

In case of Pima Diabetes and Breast Cancer, not all subsets of attributes have been tested, but only some samples with 25% and 50% of the original attributes.

Also in these cases, it seems that to less attributes correspond worst performances. There is only one exception in the breast cancer case which has approximately the same performance with the whole set of attributes and the average of 25% of missing attributes. Further, let us remark that in some cases, the accuracy of less attributes is better than the whole Information Table, for instance in Pima data sets and attributes 1 and 4 deleted.

5 Conclusion

The monotone increase of knowledge with respect to attributes in decision tables has been analyzed. From a theoretical standpoint, we saw that better approximations and classification power (in term of positive regions and decision

rules) correspond to this increase of knowledge. Algorithms to update reducts and rules have been proposed. In order to quantify the increase of rule performance, some preliminary tests have been reported. As one could expect, and with only few exceptions, the results point out that worst rules correspond to a decrease of knowledge. These results can also be useful if interpreted as a way to preprocess data. Indeed, we can have some information on which attributes should be retained and which not in order to generate rules. As pointed out in the previous section, in the Iris case we had the indication that attributes 3 and 4 have more influence on classification than attributes 1 and 2. More important, it can happen that deleting some attributes we get better performances as in the case of attributes 1,4 in the Pima case. Of course, in order to better understand when there is an increase of accuracy in deleting attributes and which is the gain, more tests are needed . It is not excluded that in this step it will be of some help to use other software besides ROSE 2 or to develop ad hoc libraries.

References

1. Cattaneo, G., Ciucci, D.: Investigation about time monotonicity of similarity and preclusive rough approximations in incomplete information systems. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 38–48. Springer, Heidelberg (2004)
2. Ciucci, D.: Classification of dynamics in rough sets. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 257–266. Springer, Heidelberg (2010)
3. Deng, D., Huang, H.-K.: Dynamic reduction based on rough sets in incomplete decision systems. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 76–83. Springer, Heidelberg (2007)
4. Grzymala-Busse, J., Grzymala-Busse, W.: Inducing better rule sets by adding missing attribute values. In: Chan, C., Grzymala-Busse, J., Ziarko, W. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 160–169. Springer, Heidelberg (2008)
5. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)
6. Moshkov, M.J., Skowron, A., Suraj, Z.: Extracting relevant information about reduct sets from data tables. In: Transactions on Rough Sets [10], pp. 200–211
7. Pawlak, Z.: Information systems - theoretical foundations. Information Systems 6, 205–218 (1981)
8. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Information Sciences 177, 41–73 (2007)
9. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)
10. Peters, J.F., Skowron, A., Rybiński, H. (eds.): Transactions on Rough Sets IX. LNCS, vol. 5390. Springer, Heidelberg (2008)
11. Shan, N., Ziarko, W.: Data-based acquisition and incremental modification of classification. Computational Intelligence 11, 357–370 (1995)
12. Swiniarski, R.W., Pancerz, K., Suraj, Z.: Prediction of model changes of concurrent systems described by temporal information systems. In: Arabnia, H.R., Scime, A. (eds.) Proceedings of The 2005 International Conference on Data Mining, DMIN 2005, Las Vegas, Nevada, USA, June 20-23 , pp. 51–57. CSREA Press (2005)

A Comparison of Some Rough Set Approaches to Mining Symbolic Data with Missing Attribute Values

Jerzy W. Grzymala-Busse[1,2]

[1] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA
[2] Institute of Computer Science, Polish Academy of Sciences, 01–237 Warsaw, Poland
jerzy@ku.edu

Abstract. This paper presents results of experiments on incomplete data sets obtained by random replacement of attribute values with symbols of missing attribute values. Rule sets were induced from such data using two different types of lower and upper approximation: local and global, and two different interpretations of missing attribute values: lost values and "do not care" conditions. Additionally, we used a probabilistic option, one of the most successful traditional methods to handle missing attribute values. In our experiments we recorded the total error rate, a result of ten-fold cross validation. Using the Wicoxon matched-pairs signed ranks test (5% level of significance for two-tailed test) we observed that for missing attribute values interpreted as "do not care" conditions, the global type of approximations is worse than the local type and that the probabilistic option is worse than the local type.

1 Introduction

Many real-life data sets are incomplete. In some cases an attribute value was accidentally erased or is unreadable. The most cautious approach to missing attribute values of this type is mining data using only specified attribute values. This type of missing attribute values will be called *lost* and denoted by "?".

Another type of missing attribute values may be exemplified by reluctance to answer some questions. For example, a patient is tested for flu and one of the questions is a color of hair. This type of missing attribute values will be called a *"do not care" condition* and denoted by "*".

We studied data sets with all missing attribute values lost, using rough set approach, for the first time in [1]. In this paper two algorithms for rule induction from such data were presented. The same data sets were studied later, see, e.g., [2,3].

The first study of "do not care" conditions, again using rough set theory, was presented in [4], where a method for rule induction in which missing attribute values were replaced by all values from the domain of the attribute was introduced. "Do not care" conditions were also studied later, see, e.g. [5,6].

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 52–61, 2011.

In our experiments we used two types of approximations: local and global. The idea of local approximations was introduced in [7]. The results of experiments [8] show superiority of local approximations. This paper presents new results of extensive experiments on symbolic data with missing attribute values. For missing attribute values interpreted as "do not care" conditions, for which local approximations are not reduced to global approximations [9], the results of experiments show that local approximations are better than global ones. Additionally, local approximations are better than a probabilistic option, one of the most successful, traditional imputation method handling missing attribute values based on the largest conditional probability given the concept to which the case belongs. This option was included to our experiments for completeness.

2 Rough Set Approaches to Missing Attribute Values

An important tool to analyze data sets is a *block of an attribute-value pair*. Let (a, v) be an attribute-value pair. For *complete* decision tables, i.e., decision tables in which every attribute value is specified, a block of (a, v), denoted by $[(a, v)]$, is the set of all cases x for which $a(x) = v$, where $a(x)$ denotes the value of the attribute a for the case x. For incomplete decision tables the definition of a block of an attribute-value pair is modified.

- If for an attribute a there exists a case x such that $a(x) = ?$, i.e., the corresponding value is lost, then the case x should not be included in any blocks $[(a, v)]$ for all values v of attribute a,
- If for an attribute a there exists a case x such that the corresponding value is a "do not care" condition, i.e., $a(x) = *$, then the case x should be included in blocks $[(a, v)]$ for all specified values v of attribute a.

For a case $x \in U$ the *characteristic set* $K_B(x)$ is defined as the intersection of the sets $K(x, a)$, for all $a \in B$, where the set $K(x, a)$ is defined in the following way:

- If $a(x)$ is specified, then $K(x, a)$ is the block $[(a, a(x)]$ of attribute a and its value $a(x)$,
- If $a(x)) =?$ or $a(x) = *$ then the set $K(x, a) = U$.

Note that for incomplete data there is a few possible ways to define approximations [8,10], we used *concept* approximations. A B-*global lower approximation* of the concept X is defined as follows:

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

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

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

A set T of attribute-value pairs, where all attributes belong to the set B and are distinct, will be called a *B-complex*. A B-*local lower* approximation of the concept X is defined as follows

$$\cup\{[T] \mid T \text{ is a B-complex of } X,\ [T] \subseteq X\}.$$

A B-*local upper* approximation of the concept X is defined as the minimal set containing X and defined in the following way

$$\cup\{[T] \mid \exists \text{ a family } \mathcal{T} \text{ of B-complexes of } X \text{ with } \forall\ T \in \mathcal{T},\ [T] \cap X \neq \emptyset\}.$$

For rule induction from incomplete data we used the MLEM2 data mining algorithm, for details see [11]. We used rough set methodology [12], i.e., for a given interpretation of missing attribute vales and for a particular version of the definition (local or global), *lower* and *upper approximations* were computed for all concepts and then rule sets were induced, *certain* rules from lower approximations and *possible* rules from upper approximations.

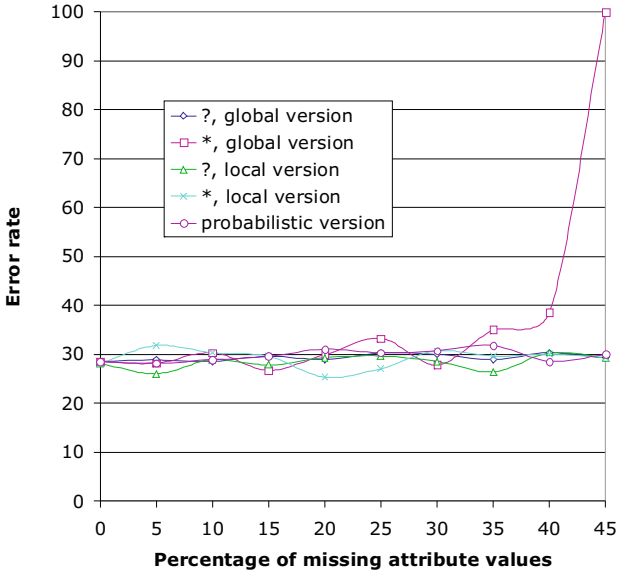


Fig. 1. Results of experiments on breast cancer data set, certain rule set

3 A Probabilistic Option

In our experiments we used a probabilistic method to handle missing attribute values based on conditional probability. This method is frequently used in data mining and is considered to be very successful. There are many other methods to deal with missing attribute values, see, e.g., [13].

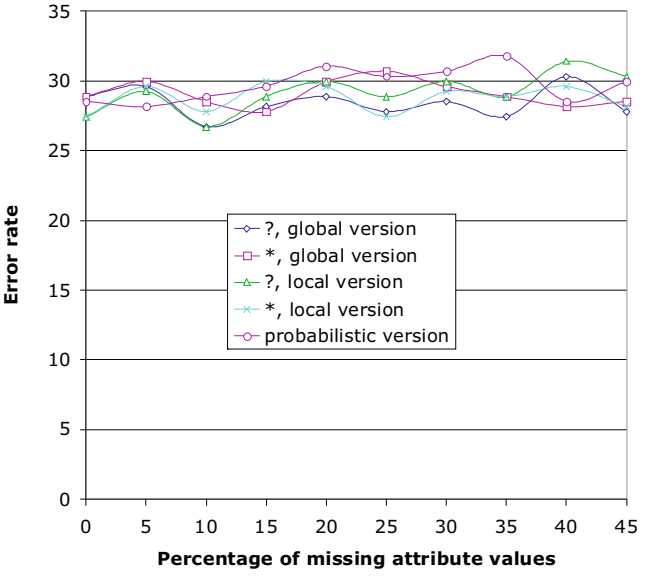


Fig. 2. Results of experiments on breast cancer data set, possible rule set

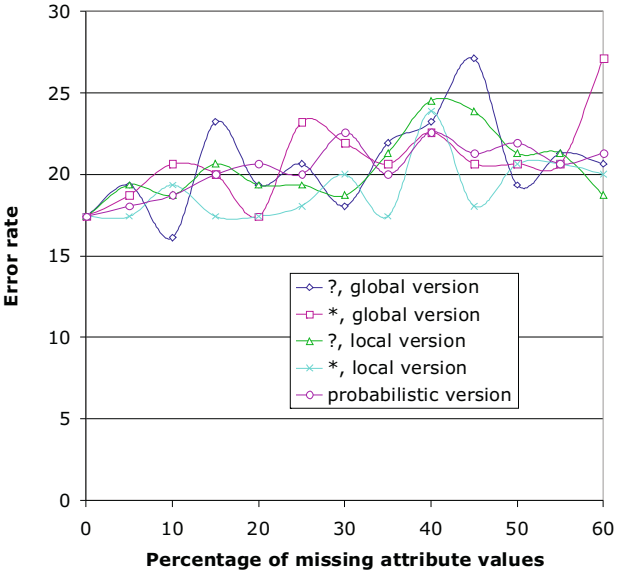


Fig. 3. Results of experiments on hepatitis data set, certain rule set

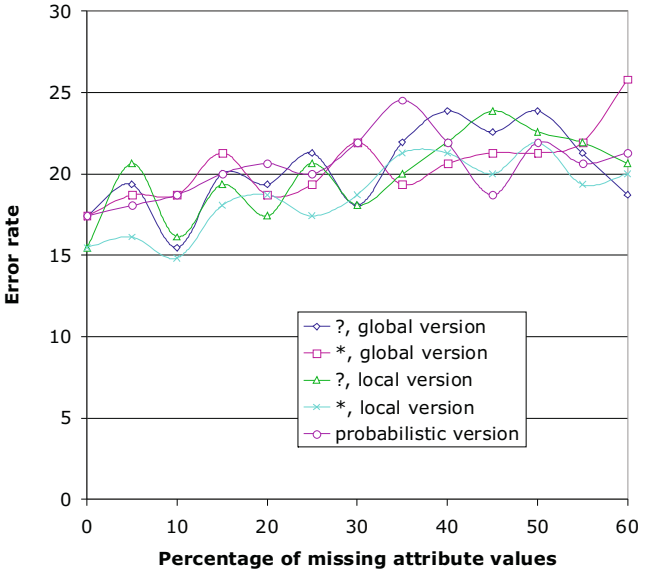


Fig. 4. Results of experiments on hepatitis data set, possible rule set

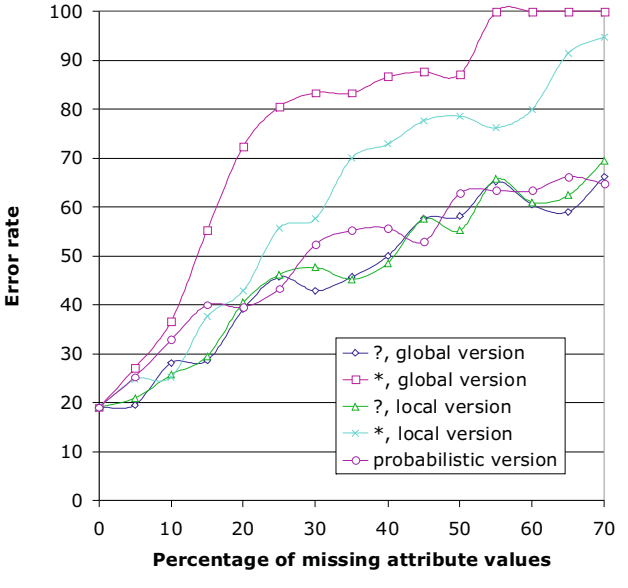


Fig. 5. Results of experiments on image data set, certain rule set

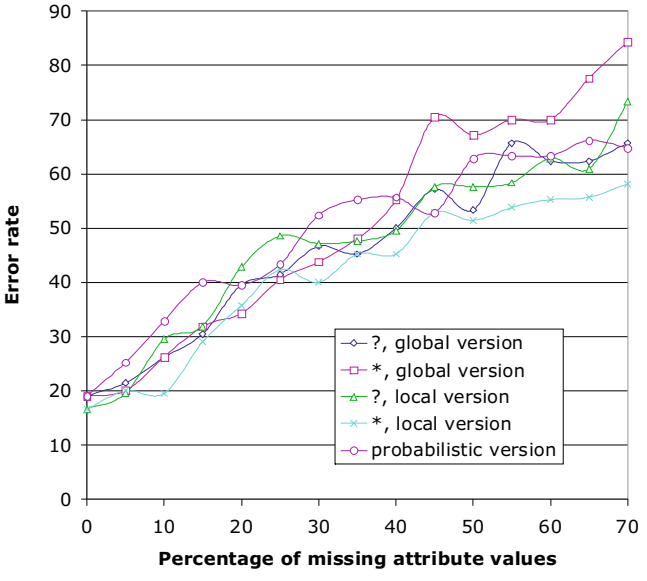


Fig. 6. Results of experiments on image data set, possible rule set

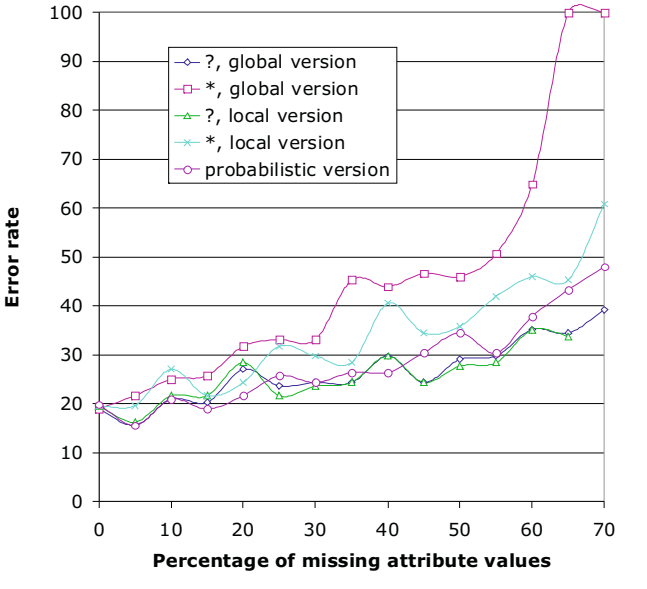


Fig. 7. Results of experiments on lymphography data set, certain rule set

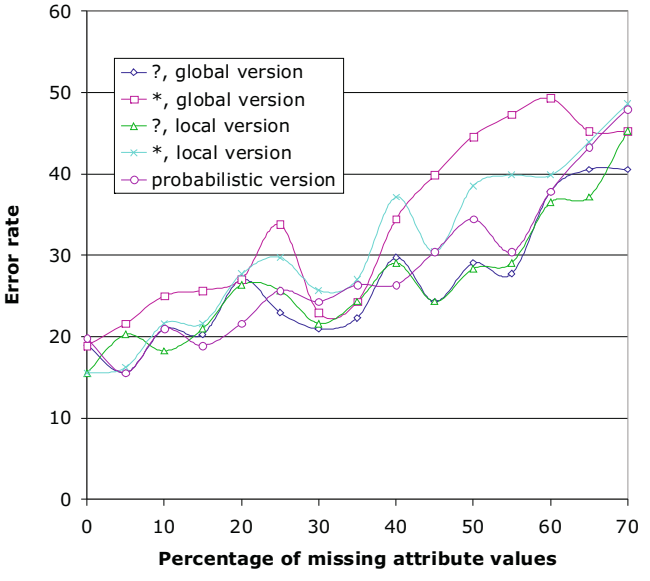


Fig. 8. Results of experiments on lymphography data set, possible rule set

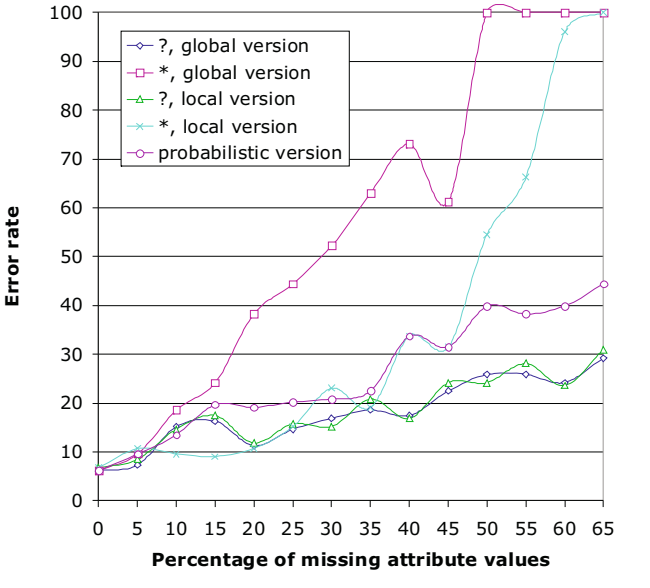


Fig. 9. Results of experiments on wine data set, certain rule set

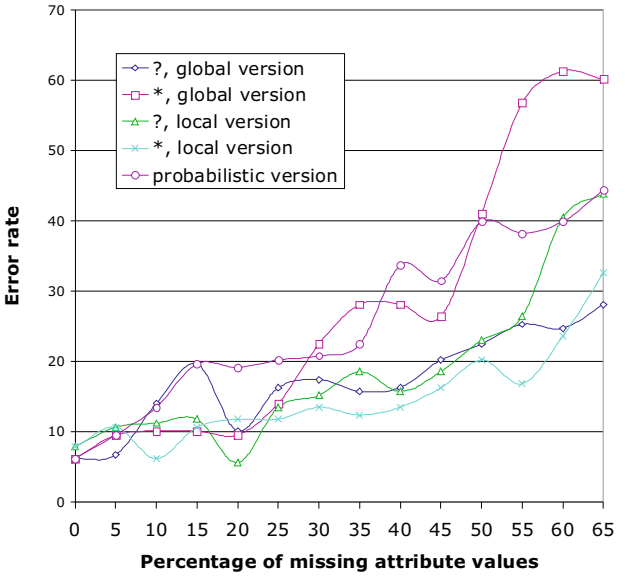


Fig. 10. Results of experiments on wine data set, possible rule set

In this method, for any case x, a missing attribute value was replaced by the most common value of the same attribute restricted to the concept to which x belongs. Thus, a missing attribute value was replaced by an attribute value with the largest conditional probability, given the concept to which the case belongs.

4 Experiments

In our experiments we used five well-known data sets, accessible at the University of California at Irvine Data Depository. Three data sets: *hepatitis*, *image segmentation* and *wine*, were discretized using a discretization method based on cluster analysis [14].

For every data set a set of templates was created. Templates were formed by replacing incrementally (with 5% increment) existing specified attribute values by *lost values*. Thus, we started each series of experiments with no *lost values*, then we added 5% of *lost values*, then we added additional 5% of *lost values*, etc., until at least one entire row of the data sets was full of *lost values*. Then three attempts were made to change configuration of new *lost values* and either a new data set with extra 5% of *lost values* was created or the process was terminated. Additionally, the same formed templates were edited for further experiments by replacing question marks, representing *lost values*, by "*"s, representing *"do not care" conditions*.

For each data set with some percentage of missing attribute values of a given type, experiments were conducted separately for certain and possible rule sets,

using four rough set approaches and the probabilistic option, respectively. Ten-fold cross validation was used to compute an error rate. Rule sets were induced by the MLEM2 option of the LERS data mining system. Results of our experiments are presented in Figures 1–10.

5 Conclusions

As follows from our experiments, there is not much difference in performance between certain and possible rule sets, with two exceptions: the certain rule set induced by the global version of MLEM2 combined with the "do not care" condition interpretation of missing attribute values is the worst approach overall. Using the Wicoxon matched-pairs signed ranks test (5% level of significance for two-tailed test) we observed that for missing attribute values interpreted as "do not care" conditions, the global version is worse than the local version, for both certain and possible rule sets (this observation is backed up by all data sets except *breast cancer* where the outcome is indecisive). Note that for the local and global options, for missing attribute values interpreted as *lost* values, for both certain and possible rule sets, the outcome is indecisive. This is not difficult to explain: for *lost* values, local approximations are reduced to global approximations, as it was observed in [9]. Using the same test we draw two additional observations. The probabilistic option is worse than local option combined with missing attribute values interpreted as "do not care" conditions, for both certain and possible rule sets. This observation is supported by all data sets except the *breast cancer* for both certain and possible rule sets and *wine* data sets for certain rule sets, where the outcome is indecisive. Additionally, for some data sets and some types of rule sets (for the *image* and *wine* data sets combined with certain rule sets and for the *lymphography* data set, for both certain and possible rule sets), the global option combined with missing attribute values interpreted as "do not care" conditions is worse than the probabilistic option. For remaining data sets and type of rule sets the outcome is indecisive.

An additional observation is related to an increase of the error rate with the increase of the percentage of missing attribute values. For some data sets (e.g., the *breast cancer* data set) the error rate is more or less constant. For some data sets (e.g., the *hepatitis* data set) the increase of the error rate is not essential. For some data sets the increase of the error rate is obvious (remaining three data sets).

The difference in performance between rough set approaches and probabilistic versions are frequently substantial. Our final conclusion is that for a given data set all ten methods should be tested and the best one should be applied for data mining.

References

1. Grzymala-Busse, J.W., Wang, A.Y.: Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. In: Proceedings of the Fifth International Workshop on Rough Sets and Soft Computing (RSSC 1997) at the Third Joint Conference on Information Sciences (JCIS 1997), pp. 69–72 (1997)

2. Stefanowski, J., Tsoukias, A.: On the extension of rough sets under incomplete information. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 73–82. Springer, Heidelberg (1999)
3. Stefanowski, J., Tsoukias, A.: Incomplete information tables and rough classification. Computational Intelligence 17(3), 545–566 (2001)
4. Grzymala-Busse, J.W.: On the unknown attribute values in learning from examples. In: Proceedings of the ISMIS 1991, 6th International Symposium on Methodologies for Intelligent Systems, pp. 368–377 (1991)
5. Kryszkiewicz, M.: Rough set approach to incomplete information systems. In: Proceedings of the Second Annual Joint Conference on Information Sciences, pp. 194–197 (1995)
6. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113(3-4), 271–292 (1999)
7. Grzymala-Busse, J.W., Rzasa, W.: Local and global approximations for incomplete data. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 244–253. Springer, Heidelberg (2006)
8. Grzymala-Busse, J.W., Rzasa, W.: A local version of the mlem2 algorithm for rule induction. Fundamenta Informaticae 100, 99–116 (2010)
9. Grzymala-Busse, J.W., Rzasa, W.: Local and global approximations for incomplete data. Transactions on Rough Sets 8, 21–34 (2008)
10. Grzymala-Busse, J.W.: Three approaches to missing attribute values—a rough set perspective. In: Proceedings of the Workshop on Foundation of Data Mining, in Conjunction with the Fourth IEEE International Conference on Data Mining, pp. 55–62 (2004)
11. Grzymala-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 243–250 (2002)
12. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
13. Grzymala-Busse, J.W., Hu, M.: A comparison of several approaches to missing attribute values in data mining. In: Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing, pp. 340–347 (2000)
14. Chmielewski, M.R., Grzymala-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. International Journal of Approximate Reasoning 15(4), 319–331 (1996)

Action Reducts

Seunghyun Im[1], Zbigniew Ras[2,3], and Li-Shiang Tsay[4]

[1] Computer Science Department, University of Pittsburgh at Johnstown, Johnstown, PA 15904, USA
sim@pitt.edu
[2] Computer Science Department, University of North Carolina, Charlotte, NC 28223, USA
[3] Institute of Computer Science, Warsaw University of Technology, 00-665 Warsaw, Poland
ras@uncc.edu
[4] ECIT Department, NC A&T State University, Greensboro, NC, 27411, USA
ltsay@ncat.edu

Abstract. An action is defined as controlling or changing some of attribute values in an information system to achieve desired result. An action reduct is a minimal set of attribute values distinguishing a favorable object from other objects. We use action reducts to formulate necessary actions. The action suggested by an action reduct induces changes of decision attribute values by changing the condition attribute values to the distinct patterns in action reducts.

Keywords: Reduct, Action Reduct, Prime Implicant, Rough Set.

1 Introduction

Suppose that the customers of a bank can be classified into several groups according to their satisfaction levels, such as satisfied, neutral, or unsatisfied. One thing the bank can do to improve the business is finding a way to make the customers more satisfied, so that they continue to do the business with the bank. The algorithm described in this paper tries to solve such problem using existing data. Assume that a bank maintains a database for customer information in a table. The table has a number of columns describing the characteristics of the customers, such as personal information, account data, survey result etc. We divide the customers into two groups based on the satisfaction level (decision value). The first group is comprised of satisfied customers who will most likely keep their account active for an extended period of time. The second group is comprised of neutral or unsatisfied customers. We find a set of distinct values or unique patterns from the first group that does not exist in the second group. The unique characteristics of the satisfied customers can be used by the bank to improve the customer satisfaction for the people in the second group. Clearly, some of attribute values describing the customers can be controlled or changed, which is defined as an action. In this paper, we propose the concept of action reduct to formulate necessary actions. An action reduct has following properties; (1) It is obtained from objects having favorable decision values. (2) It is a distinct set of values not found in the other group, the group not having favorable decision values. (3) It is the minimal set of differences between separate groups of objects. The minimal set has the advantages when formulating actions because smaller changes are easier to undertake.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 62–69, 2011.

The rest of this paper is organized as follows. Chapter 2 describes the algorithm. Related works are presented in Chapter 3. Implementation and experimental results are shown in Chapter 4. Chapter 5 concludes the paper.

Table 1. Information System S. The decision attribute is D. The condition attributes are B, C, and E. There are eight objects referred as $x_1 \sim x_8$

	B	C	E	D
x_1	b_2	c_1	e_1	d_2
x_2	b_1	c_3	e_2	d_2
x_3	b_1	c_1		d_2
x_4	b_1	c_3	e_1	d_2
x_5	b_1	c_1	e_1	d_1
x_6	b_1	c_1	e_1	d_1
x_7	b_2		e_2	d_1
x_8	b_1	c_2	e_2	d_1

2 Algorithm

2.1 Notations

We will use the following notations throughout the paper.

By an information system [2] we mean a triple $S=(X,A,V)$, where

$X = \{x_1,x_2,...x_i\}$ *is a finite set of objects,*
$A = \{a_1,a_2,...a_j\}$ *is a finite set of attributes, defined as partial functions from X into V,*
$V = \{v_1,v_2,...,v_k\}$ *is a finite set of attribute values.*

We also assume that $V = \cup\{V_a : a \in A\}$, where V_a is a domain of attribute a.

For instance, in the information system S presented by Table 1, x_1 refers to the first row and $B(x_1) = b_2$. There are four attributes, $A = \{B, C, E, D\}$. We classify the attributes into two types: condition and decision. The condition attributes are B, C, and E, and the decision attribute is D. We assume that the set of condition attributes is further partitioned into stable attributes, A_S, and flexible attributes, A_F. An attribute is called stable if the values assigned to objects do not change over time. Otherwise, the attribute is flexible. Birth date is an example of a stable attribute. Interest rate is a flexible attribute.

$A_F = \{B,C\}$
$A_S = \{E\}$
$D =$ *decision attribute*

The values in D are divided into two sets.

$d_\alpha = \{ v_i \in V_D;$ v_i *is a desired decision value}*
$d_\beta = V_D - d_\alpha$

For simplicity of presentation, we use an example having only one element d_2 in d_α and d_1 in d_β ($d_\alpha = \{d_2\}$, $d_\beta = \{d_1\}$). However, the algorithm described in the next section directly carries over to the general case where $|d_\alpha| \geq 2$ and $|d_\beta| \geq 2$.

The objects are partitioned into 2 groups based on the decision values.

$X_\alpha = \{x_i \in X; D(x_i) \in d_\alpha\}$; *i.e., objects that the decision values are in* d_α
$X_\beta = \{x_j \in X; D(x_j) \in d_\beta\}$; *i.e., objects that the decision values are in* d_β

In Table 1, $X_\alpha = \{x_1, x_2, x_3, x_4\}$ and $X_\beta = \{x_5, x_6, x_7, x_8\}$.

2.2 Algorithm Description

We want to provide the user a list of attribute values that can be used to make changes on some of the objects to steer the unfavorable decision value to a more favorable value. We use the reduct [2][9] to create that list.

By a reduct relative to an object x we mean a minimal set of attribute values distinguishing x from all other objects in the information system. For example, $\{b_2, e_2\}$ is a reduct relative to x_7 since $\{b_2, e_2\}$ can differentiate x_7 from other objects in S (see Table 1).

Now, we will extend the concept of "reduct relative to an object" to "α-reduct". We partitioned the objects in S into two groups by their decision values. Objects in X_α have d_2 that is the favorable decision value. Our goal is to identify which sets of condition attribute values describing objects in X_α make them different from the objects in X_β. Although there are several different ways to find distinct condition attribute values (e.g. association rules), reducts have clear advantages; it does not require the user to specify the rule extraction criteria, such as support and confidence values, while generating the minimal set of distinct sets. Thereby, the algorithm is much easier to use, and creates a consistent result across different users.

Table 5 shows α-reducts for S. Those are the smallest sets of condition attribute values that are different from the condition attribute values representing objects in X_β. We obtained the first two α-reducts relative to x_1. These are the prime implicants [1] (see the next section for details) of the differences between x_1 and $\{x_5, x_6, x_7, x_8\}$. Subsequent α-reducts are extracted using objects x_2, x_3, and x_4.

We need a method to measure the usability of the α-reducts. Two factors, *frequency* and *hit ratio*, determine the usability. (1) Frequency : More than one object can have the same α-reduct. The *frequency* of an α-reduct in X_α is denoted by f. (2) Hit Ratio : The *hit ratio*, represented as h, is the ratio between the number of applicable objects and the total number of objects in X_β. An applicable object is the object that the attribute values are different from those in α-reduct, and they are not stable values. The α-reducts may not be used to make changes for some of the attribute values of $x \in X_\beta$ for two reasons. Some objects in X_β do not differ in terms of their attribute values. Therefore, changes cannot be made. Second, we cannot modify stable attribute values. It does not make sense to suggest a change of un-modifiable values. We define the following function to measure the usability. The weight of an α-reduct k is,

$$w_k = (f_k \cdot h_k) / (\Sigma(f \cdot h)),$$

where f_k and h_k are the frequency and hit ratio for k, and $\Sigma(f \cdot h)$ is the sum of the weights of all α-reducts. It provides a way to prioritize the α-reduct using a normalized value.

Table 2. X_α. The objects classified as d_2. We assume that d_2 is the favorable decision.

	B	C	E	D
x_1	b_2	c_1	e_1	d_2
x_2	b_1	c_3	e_2	d_2
x_3	b_1	c_1		d_2
x_4	b_1	c_3	e_1	d_2

Table 3. X_β. The objects classified as d_1

	B	C	E	D
x_5	b_1	c_1	e_1	d_1
x_6	b_1	c_1	e_1	d_1
x_7	b_2		e_2	d_1
x_8	b_1	c_2	e_2	d_1

2.3 Example

Step 1. Finding α-reducts

Using the partitions in Tables 2 and 3, we find distinct attribute values of $x \in X_\alpha$ against $x \in X_\beta$. The following matrix shows the discernable attribute values for $\{x_1, x_2, x_3, x_4\}$ against $\{x_5, x_6, x_7, x_8\}$

Table 4. Discernable attribute values for $\{x_1, x_2, x_3, x_4\}$ against $\{x_5, x_6, x_7, x_8\}$

	x_1	x_2	x_3	x_4
x_5	b_2	$c_3 + e_2$		c_3
x_6	b_2	$c_3 + e_2$		c_3
x_7	$c_1 + e_1$	$b_1 + c_3$	$b_1 + c_1$	$b_1 + c_3 + e_1$
x_8	$b_2 + c_1 + e_1$	c_3	c_1	$c_3 + e_1$

For example, b_2 in x_1 is different from x_5. We need b_2 to discern x_1 from x_5. Either c_1 or (or is denoted as + sign) e_1 can be used to distinguish x_1 from x_7. In order to find the *minimal* set of values that distinguishes x_1 from all objects in $X_\beta = \{x_5, x_6, x_7, x_8\}$ we multiply all discernable values: $(b_2) \times (b_2) \times (c_1 + e_1) \times (b_2 + c_1 + e_1)$. That is, (b_2) and $(c_1$ or $e_1)$ and $(b_2$ or c_1 or $e_1)$ should be different to make x_1 distinct from all other

objects. The process is known as finding a prime implicant by converting the conjunction normal form (CNF) to disjunction normal form (DNF)[2][9]. The α-reduct $r(x_1)$ relative to x_1 is computed using the conversion and the absorption laws:

$$\begin{aligned} r(x_1) &= (b_2) \times (b_2) \times (c_1 + e_1) \times (b_2 + c_1 + e_1) \\ &= (b_2) \times (b_2) \times (c_1 + e_1) \\ &= (b_2 \times c_1) + (b_2 \times e_1) \end{aligned}$$

A missing attribute value of an object in X_α does not qualify to discern the object from the objects in X_β because it is undefined. A missing value in X_β, however, is regarded as a different value if a value is present in $x \in X_\alpha$. When a discernible value does not exist, we do not include it in the calculation of the prime implicant. We acquired the following α-reducts:

$r(x_1) = (b_2 \times c_1) + (b_2 \times e_1)$
$r(x_2) = (c_3)$
$r(x_3) = (c_1)$
$r(x_4) = (c_3)$

Step 2. Measuring the usability of α*-reduct*

Table 5 shows the α-reducts for information System S. The frequency of α-reduct $\{b_2, c_1\}$ is 1 because it appears in X_α once. The hit ratio is *4/4* = 1, meaning that we can use the reduct for all objects in X_β. The weight is 0.25, which is acquired by dividing its weight, $f \cdot h = 1$, by the sum of all weights, $\Sigma(f \cdot h) = (1 \cdot 1) + (1 \cdot 0.5) + (2 \cdot 1) + (1 \cdot 0.5) = 4$.

The values in the stable attribute *E* cannot be modified. Therefore, the hit ratio for α-reduct $\{b_2, e_1\}$ is *2/4* = 0.5 because the stable value, e_2, in x_7 and x_8 cannot be converted to e_1.

Table 5. α-reduct for Information Sytem S. * indicate a stable attribute value.

α-reduct	Weight (*w*)	Frequency (*f*)	Hit Ratio (*h*)
$\{b_2, c_1\}$	*0.25 (25%)*	*1*	*1*
$\{b_2, e_1^*\}$	*0.125 (12.5%)*	*1*	*0.5*
$\{c_3\}$	*0.5 (50%)*	*2*	*1*
$\{c_1\}$	*0.125 (12.5%)*	*1*	*0.5*

Using the α-reduct $\{c_3\}$ that has the highest weight, we can make a recommendation; change the value of *C* in X_β to c_3 in order to induce the decision value in X_β to d_2.

3 Implementation and Experiment

We implemented the algorithm in Python 2.6 on a MacBook computer running OS X, and tested it using a sample data set (lenses) obtained from [8]. The data set contains

information for fitting contact lenses. Table 6 shows the attributes names, descriptions, and the partitions. The decision attribute, lenses, has three classes. We set the second class (soft lenses) as the favorable decision value. The data set has 4 condition attributes. We assumed that *the age of the patient* is a stable attribute, and *prescription*, *astigmatic* and *tear production rate* are flexible attributes. The favorable decision value and the attribute partition are defined only for this experiment. Their actual definitions might be different. All attributes are categorical in the dataset.

Table 6. Dataset used for the experiment

Attribute	Values	Type
(1) age of the patient *	young, pre-presbyopic, presbyopic	Stable
(2) spectacle prescription	myope, hypermetrope	Flexible
(3) astigmatic	no, yes	Flexible
(4) tear production rate	reduced, normal	Flexible
(5) lenses	the patient fitted with hard contact lenses the patient fitted with soft contact lenses = $d\alpha$ the patient not be fitted with contact lenses.	Decision

Table 7 shows the α-reducts generated during experiment. We interpret them as action reducts. For instance, the second α-reduct can be read as, change the values in attribute 2, 3, and 4 to the suggested values (hypermetrope, no, normal) in order to change the decision to 'soft lenses'. Because there is no stable attribute value in the α-reduct and the same pattern has not been found in X_β, its hit ratio is 1.

Table 7. α-reduct for d_α = soft lenses. * indicate the attribute value is stable. The number in the () is the attribute number in Table 6.

α-reduct.	Weight (*w*)	Frequency (*f*)	Hit Ratio (*h*)
young(1)*, no(3), normal(4)	*0.14*	*2*	*0.31*
hypermetrope(2), no(3), normal(4)	*0.72*	*3*	*1*
pre-presbyopic(1)*, no(3), normal(4)	*0.14*	*2*	*0.31*

4 Related Work and Contribution

The procedure for formulating an action from existing database has been discussed in many literatures. A definition of an action as a form of a rule was given in [4]. The method of action rules discovery from certain pairs of association rules was proposed in [3]. A concept similar to action rules known as interventions was introduced in [5]. The action rules introduced in [3] has been investigated further. In [12], authors present a new agglomerative strategy for constructing action rules from single classification rules. Algorithm ARAS, proposed in [13], is also an agglomerative type strategy generating action rules. The method generates sets of terms (built from values of attributes) around classification rules and constructs action rules directly from them. In [10], authors proposed a method that extracts action rules directly from attribute values from an incomplete information system without using pre-existing conditional

rules. In these earlier works, action rules are constructed from classification rules. This means that they use pre-existing classification rules or generate rules using a rule discovery algorithm, such as LERS [6], then, construct action rules either from certain pairs of the rules or from a single classification rule. The methods in [10], [13], [14] and [7] do not formulate actions directly from existing classification rules. However, the extraction of classification rules during the formulation of an action rule is inevitable because actions are built as the effect of possible changes in different rules. Action rules constructed from classification rules provide a complete list of actions to be taken. However, we want to develop a method that provides a simple set of attribute values to be modified without using a typical action rule form (e.g. [condition1→condition2]=>[decision1→decision2]) for decision makers who want to have simple recommendations. The recommendations made by α-reduct is typically quite simple, *i.e.* change a couple of values to have a better outcome. In addition, it does not require from the user to define two sets of support and confidence values; one set for classification rules, and other for action rules.

5 Summary

This paper discussed an algorithm for finding α–reducts (also called action reducts) from an information system, and presented an experimental result. An action reduct is a minimal set of attribute values distinguishing a favorable object from other objects, and are used to formulate necessary actions. The action suggested by an action reduct aims to induce the change of the decision attribute value by changing the condition attribute values to the unique pattern in the action reduct. The algorithm is developed as part of on-going research project seeking solution to the problem of reducing college freshman dropouts. We plan to run the algorithm using real world data in the near future.

Acknowledgments. This research has been partially supported by the President's Mentorship Fund at the University of Pittsburgh Johnstown.

References

1. Gimpel, J.F.: A Reduction Technique for Prime Implicant Tables. In: The Fifth Annual Symposium on Switching Circuit Theory and Logical Design, pp. 183–191. IEEE, Washington (1964)
2. Pawlak, Z.: Rough Sets-Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
3. Raś, Z.W., Wieczorkowska, A.A.: Action-Rules: How to Increase Profit of a Company. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 587–592. Springer, Heidelberg (2000)
4. Geffner, H., Wainer, J.: Modeling Action, Knowledge and Control. ECAI, 532–536 (1998)
5. Greco, S., Matarazzo, B., Pappalardo, N., Slowinski, R.: Measuring Expected Effects of Interventions based on Decision Rules. Journal of Experimental and Theoretical Artificial Intelligence 17(1-2), 103–118 (2005)

6. Grzymala-Busse, J.: A New Version of the Rule Induction System LERS. Fundamenta Informaticae 31(1), 27–39 (1997)
7. He, Z., Xu, X., Deng, S., Ma, R.: Mining Action Rules from Scratch. Expert Systems with Applications 29(3), 691–699 (2005)
8. Hettich, S., Blake, C.L., Merz, C.J. (eds.): UCI Repository of Machine Learning Databases. University of California, Dept. of Information and Computer Sciences, Irvine (1998), http://www.ics.uci.edu/mlearn/MLRepository.html
9. Skowron, A.: Rough Sets and Boolean Reasoning. In: Pedrycz, W. (ed.) Granular Computing: An Emerging Paradigm, pp. 95–124. Springer, Heidelberg (2001)
10. Im, S., Ras, Z.W., Wasyluk, H.: Action Rule Discovery from Incomplete Data. Knowledge and Information Systems 25(1), 21–33 (2010)
11. Qiao, Y., Zhong, K., Wangand, H., Li, X.: Developing Event-Condition-Action Rules in Real-time Active Database. In: ACM Symposium on Applied Computing 2007, pp. 511–516. ACM, New York (2007)
12. Raś, Z.W., Dardzińska, A.: Action Rules Discovery, a New Simplified Strategy. In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) ISMIS 2006. LNCS (LNAI), vol. 4203, pp. 445–453. Springer, Heidelberg (2006)
13. Ras, Z.W., Tzacheva, A., Tsay, L., Gurdal, O.: Mining for Interesting Action Rules. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 187–193. IEEE, Washington (2005)
14. Raś, Z.W., Dardzińska, A.: Action Rules Discovery Based on Tree Classifiers and Meta-actions. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS(LNAI), vol. 5722, pp. 66–75. Springer, Heidelberg (2009)

Incremental Rule Induction Based on Rough Set Theory

Shusaku Tsumoto

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

Abstract. Extending the concepts of rule induction methods based on rough set theory, we introduce a new approach to knowledge acquistion, which induces probabilistic rules in an incremental way, which is called PRIMEROSE-INC (Probabilistic Rule Induction Method based on Rough Sets for Incremental Learning Methods). This method first uses coverage rather than accuracy, to search for the candidates of rules, and secondly uses accuracy to select from the candidates. This system was evaluated on clinical databases on headache and meningitis. The results show that PRIMEROSE-INC induces the same rules as those induced by the former system: PRIMEROSE, which extracts rules from all the datasets, but that the former method requires much computational resources than the latter approach.

1 Introduction

There have been proposed several symbolic inductive learning methods, such as induction of decision trees [1,5], and AQ family [3]. These methods are applied to discover meaningful knowledge from large databases, and their usefulness is in some aspects ensured. However, most of the approaches induces rules from all the data in databases, and cannot induce incrementally when new samples are derived. Thus, we have to apply rule induction methods again to the databases when such new samples are given, which causes the computational complexity to be expensive even if the complexity is n^2.

Thus, it is important to develop incremental learning systems in order to manage large databases [6,9]. However, most of the previously introduced learning systems have the following two problems: first, those systems do not outperform ordinary learning systems, such as AQ15 [3], C4.5 [5] and CN2 [2]. Secondly, those incremental learning systems mainly induce deterministic rules. Therefore, it is indispensable to develop incremental learning systems which induce probabilistic rules to solve the above two problems.

Extending the concepts of rule induction methods based on rough set theory, we introduce a new approach to knowledge acquistion, which induces probabilistic rules incrementally, called PRIMEROSE-INC (Probabilistic Rule Induction Method based on Rough Sets for Incremental Learning Methods).

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 70–79, 2011.

Although the formerly proposed rule induction method PRIMEROSE [8], which extracts rules from all the data in database uses apparent accuracy to search for probabilistic rules, PRIMEROSE-INC first uses coverage, to search for the candidates of rules, and secondly uses accuracy to select from the candidates. When new examples are added, firstly, the method revise the coverage and an accuracy of each elementary attribute value pairs. Then, for each pair, if coverage value decreases, then it stores it into a removal-candidate list. Else, it stores it into an acceptance-candidate list. Thirdly, for each pair in the removal candidate list, the method searches for a rule including this paer and check whether the accuracy and coverage are larger than given thresholds. Then, the same process is applied to each pair in the acceptance-candidate list. For other rules, the method revises accuracy and coverage.

This system was evaluated on two clinical databases: databases on meningoencephalitis and databases on headache with respect to the following four points: accuracy of classification, the number of generated rules, spatial computational complexity, and temporal computational complexity. The results show that PRIMEROSE-INC induced the same rules as those induced by the former system: PRIMEROSE.

2 Rough Sets and Probabilistic Rules

2.1 Rough Set Theory

Rough set theory clarifies set-theoretic characteristics of the classes over combinatorial patterns of the attributes, which are precisely discussed by Pawlak [4]. This theory can be used to acquire some sets of attributes for classification and can also evaluate how precisely the attributes of database are able to classify data.

Table 1. An Example of Database

No.	loc	nat	his	nau	class
1	who	per	per	no	m.c.h.
2	who	per	per	no	m.c.h.
3	lat	thr	per	no	migraine
4	who	thr	per	yes	migraine
5	who	per	per	no	psycho

Let us illustrate the main concepts of rough sets which are needed for our formulation. Table 1 is a small example of database which collects the patients who complained of headache.[1] First, let us consider how an attribute "loc" classify the headache patients' set of the table. The set whose value of the attribute

[1] The abbreviated attribute names of this table stand for the following: loc: location, nat: nature, his: history, nau: nausea, who: whole, lat: lateral, per: persistent, thr: throbbing, m.c.h.: muscle contraction headache, migraine: classic migraine, psycho: psychogenic headache.

"loc" is equal to "who" is {1,2,4,5}, which shows that the 1st, 2nd, 4th, 5th case (In the following, the numbers in a set are used to represent each record number). This set means that we cannot classify {1,2,4,5} further solely by using the constraint $R = [loc = who]$. This set is defined as the indiscernible set over the relation R and described as follows: $[x]_R = \{1, 2, 4, 5\}$. In this set, {1,2} suffer from muscle contraction headache("m.c.h."), {4} from classical migraine("migraine"), and {5} from psychological headache("psycho"). Hence we need other additional attributes to discriminate between m.c.h., migraine, and psycho. Using this concept, we can evaluate the classification power of each attribute. For example, "nat=thr" is specific to the case of classic migraine (migraine). We can also extend this indiscernible relation to multivariate cases, such as $[x]_{[loc=who]\wedge[nau=no]} = \{1, 2\}$ and $[x]_{[loc=who]\vee[nat=no]} = \{1, 2, 4, 5\}$, where $\wedge$ and $\vee$ denote "and" and "or" respectively. In the framework of rough set theory, the set {1,2} is called *strictly definable* by the former conjunction, and also called *roughly definable* by the latter disjunctive formula. Therefore, the classification of training samples D can be viewed as a search for the best set $[x]_R$ which is supported by the relation R. In this way, we can define the characteristics of classification in the set-theoretic framework. For example, accuracy and coverage, or true positive rate can be defined as:

$$\alpha_R(D) = \frac{|[x]_R \cap D|}{|[x]_R|}, \text{ and } \kappa_R(D) = \frac{|[x]_R \cap D|}{|D|},$$

where $|A|$ denotes the cardinality of a set A, $\alpha_R(D)$ denotes an accuracy of R as to classification of D, and $\kappa_R(D)$ denotes a coverage, or a true positive rate of R to D, respectively. For example, when R and D are set to $[nau = yes]$ and $[class = migraine]$, $\alpha_R(D) = 1/1 = 1.0$ and $\kappa_R(D) = 1/2 = 0.50$.

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

2.2 Probabilistic Rules

The simplest probabilistic model is that which only uses classification rules which have high accuracy and high coverage.[2] This model is applicable when rules of high accuracy can be derived. Such rules can be defined as:

$$R \stackrel{\alpha,\kappa}{\rightarrow} d \text{ s.t.} \quad \begin{array}{l} R = \vee_i R_i = \vee \wedge_j [a_j = v_k], \\ \alpha_{R_i}(D) > \delta_\alpha \text{ and } \kappa_{R_i}(D) > \delta_\kappa, \end{array}$$

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

$$\begin{array}{ll} [loc = who]\&[nau = no] & \rightarrow m.c.h.\ \alpha = 2/3 = 0.67, \kappa = 1.0, \\ [nat = per] & \rightarrow m.c.h.\ \alpha = 2/3 = 0.67, \kappa = 1.0, \end{array}$$

[2] In this model, we assume that accuracy is dominant over coverage.

where δ_α and δ_κ are set to 0.5 and 0.3, respectively.

It is notable that this rule is a kind of probabilistic proposition with two statistical measures, which is one kind of an extension of Ziarko's variable precision model(VPRS) [10][3].

3 Problems in Incremental Rule Induction

The most important problem in incremental learning is that it does not always induce the same rules as those induced by ordinary learning systems[4], although an applied domain is deterministic. Furthermore, since induced results are strongly dependent on the former training samples, the tendency of overfitting is larger than the ordinary learning systems.

The most important factor of this tendency is that the revision of rules is based on the formerly induced rules, which is the best way to suppress the exhaustive use of computational resources. However, when induction of the same rules as ordinary learning methods is required, computational resources will be needed, because all the candidates of rules should be considered.

Thus, for each step, computational space for deletion of candidates and addition of candidates should be needed, which causes the computational speed of incremental learning to be slow. Moreover, in case when probabilistic rules should be induced, the situation becomes much severer, since the candidates for probabilistic rules become much larger than those for deterministic rules.

For the above example, no deterministic rule can be derived from Table 1. Then, when an additional example is given as shown in Section 4.2 (the 6th example), $[loc = lat]\&[nau = yes] \rightarrow m.c.h.$ will be calculated. However, in the case of probabilistic rules, two rule will be derived under the condition that $\delta_\alpha = 0.5$ and $\delta_\kappa = 0.3$, as shown in Section 4.2 If these thresholds are not used, induced probabilistic rules becomes much larger. Thus, there is a trade-off between the performance of incremental learning methods and its computational complexity.

In our approach, we first focus on the performance of incremental learning methods, that is, we introduce a method which induces the same rules as those derived by ordinary learning methods. Then, we estimate the effect of this induction on computational complexity.

4 An Algorithm for Incremental Learning

In order to provide the same classificatory power to incremental learning methods as ordinary learning algorithms, we introduce an incremental learning method

[3] In VPRS model, the two kinds of precision of accuracy is given, and the probabilistic proposition with accuracy and two precision conserves the characteristics of the ordinary proposition. Thus, our model is to introduce the probabilistic proposition not only with accuracy, but also with coverage.

[4] Here, ordinary learning systems denote methods that induce all rules by using all the samples.

PRIMEROSE-INC (Probabilistic Rule Induction Method based on Rough Sets for Incremental Learning Methods). PRIMEROSE-INC first measures the statistical characteristics of coverage of elementary attribute-value pairs, which corresponds to selectors. Then, it measures the statistical characteristics of accuracy of the whole pattern of attribute-value pairs observed in a dataset.

In this algorithm, we use the following characteristic of coverage.

Proposition 1 (Monotonicity of Coverage)
Let R_j denote an attribute-value pair, which is a conjunction of R_i and $[a_{i+1} = v_j]$. Then,

$$\kappa_{R_j}(D) \leq \kappa_{R_i}(D).$$

Proof.
Since $[x]_{R_j} \subseteq [x]_{R_i}$ holds, $\kappa_{R_j}(D) = \frac{|[x]_{R_j} \cap D|}{|D|} \leq \frac{|[x]_{R_i} \cap D|}{|D|} = \kappa_{R_i}(D)$. □

Furthermore, in rule induction methods, R_j is selected to satisfy $\alpha_{R_j}(D) > \alpha_{R_i}(D)$. Therefore, it is sufficient to check the behavior of coverage of elementary attribute-value pairs in order to estimate the characteristics of induced rules, while it is necessary to check the behavior of accuracy of elementary attribute-value pairs and accuracy of patterns observed in the databases in order to estimate the characteristics of induced rules.

4.1 Algorithm

From these consideration, the selection algorithm is defined as follows, where the following four lists are used. $List_1$ and $List_2$ stores an elementary relation which decrease and increase its coverage, respectively, when a new training sample is given. $List_a$ is a list of probabilistic rules which satisfy the condition on the thresholds of accuracy and coverage. Finally, $List_r$ stores a list of probabilistic rules which do not satisfy the above condition.

(1) Revise the coverage and an accuracy of each elementary attribute value pair $[a_i = v_j]$ by using a new additional sample S_k.
(2) For each pair $r_{ij} = [a_i = v_j]$, if $\kappa_{r_{ij}}$ decreases, then store it into $List_1$. Else, store it into $List_2$.
(3) For each member r_{ij} in $List_1$, Search for a rule in $List_a$ whose condition R includes r_{ij} and which satisfies $\alpha_R > \delta_\alpha$ and $\kappa_R > \delta_\kappa$. Remove it from $List_a$ and Store it into $List_r$.
(4) For each member r_{ij} in $List_2$, Search for a rule in $List_r$ whose condition R includes r_{ij}. If it satisfies $\alpha_R > \delta_\alpha$ and $\kappa_R > \delta_\kappa$, then Remove it from $List_r$ and Store it into $List_a$. Else, search for a rule which satisfies the above condition by using rule induction methods.[5]
(5) For other rules in $List_r$, revise accuracy and coverage. If a rule does not satisfy $\alpha_R > \delta_\alpha$ and $\kappa_R > \delta_\kappa$, then Remove it from $List_r$ and Store it into $List_a$.

[5] That is, it makes a conjunction of attribute-value pairs and checks whether this conjunction satisfies $\alpha_R > \delta_\alpha$ and $\kappa_R > \delta_\kappa$.

4.2 Example

For example, let us consider a case when the following new sample is provided after probabilistic rules are induced from Table 1: [6]

No.	loc	nat	his	nau	class
6	lat	thr	per	no	m.c.h.

The initial condition of this system derived by Table 1 is summarized into Table 2, and $List_a$ and $List_r$ for m.c.h. are given as follows: $List_a = \{[loc = who]\&[nau = no], [nat = per]\}$, and $List_r = \{[loc = who], [loc = lat], [nat = thr], [his = per], [nau = yes], [nau = no]\}$. Then, the first procedure revises

Table 2. Accuracy and Coverage of Elementary Relations (m.c.h.)

Relation	Accuracy	Coverage
$[loc = who]$	0.5	1.0
$[loc = lat]$	0.0	0.0
$[nat = per]$	0.67	1.0
$[nat = thr]$	0.0	0.0
$[his = per]$	0.4	1.0
$[nau = yes]$	0.0	0.0
$[nau = no]$	0.5	1.0

Table 3. Revised Accuracy and Coverage of Elementary Relations (m.c.h.)

Relation	Accuracy	Coverage
$[loc = who]$	0.5	**0.67**
$[loc = lat]$	**0.5**	**0.33**
$[nat = per]$	0.67	**0.67**
$[nat = thr]$	**0.33**	**0.33**
$[his = per]$	**0.5**	1.0
$[nau = yes]$	**0.5**	**0.33**
$[nau = no]$	0.4	**0.67**

accuracy and coverage for all the elementary relations (Table 3). Since the coverages of $[loc = lat]$, $[nat = thr]$, and $[nau = yes]$ become larger than 0.3, they are included in $List_2$. In the same way, $[loc = who]$, $[nat = per]$, and $[nau = no]$ are included in $List_1$.

Next, the second procedure revises two measures for all the rules in $List_a$ whose conditional parts include a member of $List_1$. Then, the formerly induced probabilistic rules are revised into:

[6] In this example, δ_α and δ_κ are again set to 0.5 and 0.3, respectively.

$$[loc = who]\&[nau = no] \rightarrow m.c.h.\ \alpha = 0.67,\ \kappa = 0.67,$$
$$[nat = per] \rightarrow m.c.h.\ \alpha = 0.67,\ \kappa = 0.67,$$

and none of them are not removed from $List_a$. Then, the third procedure revises two measures for all the rules in $List_r$ whose conditional parts include a member of $List_2$. Then, the following probabilistic rule satisfies $\alpha > 0.5$ and $\kappa > 0.3$:

$$[loc = lat]\&[nau = yes] \rightarrow m.c.h.\ \alpha = 1.0,\ \kappa = 0.33,$$

and is stored into $List_a$. Finally, $List_a$ and $List_r$ for m.c.h. are calculated as follows:

$$List_a = \{[loc = who]\&[nau = no], [nat = per]\&[nau = no], [loc = lat]\&[nau = yes]\},$$
$$List_r = \{[loc = who], [loc = lat], [nat = per], [nat = thr], [his = per], [nau = yes], [nau = no]\}.$$

5 Experimental Results

PRIMEROSE-INC was applied to headache and meningitis, whose precise information is given in Table 4, and compared with PRIMEROSE [8], its deterministic version PRIMEROSE0[7], C4.5, CN2 and AQ15. The experiments

Table 4. Information about Databases

Domain	Samples	Classes	Attributes
headache	1477	10	20
meningitis	198	3	25

were conducted by the following three procedures. First, these samples were randomly splits into pseudo-training samples and pseudo-test samples. Second, by using the pseudo-training samples, PRIMEROSE-INC, PRIMEROSE, and PRIMEROSE0 induced rules and the statistical measures[8]. Third, the induced results were tested by the pseudo-test samples. These procedures were repeated for 100 times and average each accuracy and the estimators for accuracy of diagnosis over 100 trials.

Table 7 and 8 give the comparison between PRIMEROSE-INC and other rule induction methods with respect to the averaged classification accuracy and the number of induced rules. These results show that PRIMEROSE-INC attains the same performance of PRIMEROSE, which is the best performance in those rule induction systems. These results show that PRIMEROSE-INC overperforms all the other non-incremental learning methods, although they need much larger memory space for running.

[7] This version is given by setting δ_α to 1.0 and δ_κ to 0.0 in PRIMEROSE.
[8] The thresholds δ_α and δ_κ is set to 0.75 and 0.5, respectively in these experiments.

6 Related Work

Shan and Ziarko [6] introduce decision matrix method, which is based on an indiscernible matrix introduced by Skowron and Rauszer [7], in order to make incremental learning methods efficient.

Their approach is simple, but very powerful. For the above example shown in Table 1, the decision matrix for m.c.h. is given as Table 5, where the rows denote positive examples of m.c.h., the columns denote negative examples of m.c.h., and each matrix element a_{ij} shows the differences in attribute-value pairs between ith sample and jth sample. Also, ϕ denotes that all the attribute-value pairs in two samples are the same. Shan and Ziarko discuss induction

Table 5. Decision Matrix for m.c.h.

U	3	4	5
1	$(l = w), (n = p)$	$(n = p), (n_a = n)$	ϕ
2	$(l = w), (n = p)$	$(n = p), (n_a = n)$	ϕ

NOTATIONS: l=w: loc=who, n=p: nat=per, $n_a = n$: nau=no.

of deterministic rules in their original paper, but it is easy to extend it into probabilistic domain. In Table 5, the appearance of ϕ shows that decision rules for m.c.h. should be probabilistic. Since the first and the second row have the same pattern, {1,2,5} have the same pattern of attribute-value pairs, whose accuracy is equal to 2/3=0.67.

Furthermore, rules are obtained as: $([loc = who] \vee [nat = per]) \wedge ([nat = per] \vee [nau = no]) \rightarrow m.c.h.$, which are exactly the same as shown in Section 4.

When a new example is given, it will be added to a row when it is a positive example, and a column when a negative example. Then, again, new matrix elements will be calculated. For the above example, the new decision matrix will be obtained as in Table 6.

Then, from the last row, the third rule $[loc = lat] \wedge [nau = yes] \rightarrow m.c.h.$ is obtained.

The main difference between our method and decision matrix is that the latter approach is based on apparent accuracy, rather than coverage. While the same results are obtained in the above simple example, the former approach is sensitive to the change of coverage and the latter is to the change of accuracy. Thus, if we need rules of high accuracy, decision matrix technique is very powerful. However, when an applied domain is not so stable, calculation of rules is not so easier. Furthermore, in such an application area, a rule of high accuracy supports only a small case, which suggests that this rule is overfitted to the training samples. Thus, in this domain, coverage should be dominant over accuracy in order to suppress the tendency of overfitting, although rules of high coverage are weaker than rules of high accuracy in general. This suggests that there is a trade-off between accuracy and coverage. As shown in the definition, the difference between accuracy and coverage is only their denominators: $[x]_R$ and D. Although

Table 6. Decision Matrix with Additional Sample

U	3	4	5
1	$(l = w), (n = p)$	$(n = p), (n_a = n)$	ϕ
2	$(l = w), (n = p)$	$(n = p), (n_a = n)$	ϕ
6	$(na = y)$	$(l = l)$	$(l = l), (n = t), (n_a = y)$

NOTATIONS: l=w: loc=who, l=l: loc=lat, n=p: nat=per, n=t: nat=thr, $n_a = y/n$: nau=yes/no

it is difficult to discuss the differences in the behavior between accuracy and coverage, accuracy is inversely proportional to coverage in many domains.

However, original decision matrix technique does not incorporate such calculation of coverage. Thus, it needs to include such calculation mechanism when we extend it into the usage of both statistical measures.

Table 7. Experimental Results: Accuracy and Number of Rules (Headache)

Method	Accuracy	No. of Rules
PRIMEROSE-INC	$89.5 \pm 5.4\%$	67.3 ± 3.0
PRIMEROSE	$89.5 \pm 5.4\%$	67.3 ± 3.0
PRIMEROSE0	$76.1 \pm 1.7\%$	15.9 ± 4.1
C4.5	$85.8 \pm 2.4\%$	16.3 ± 2.1
CN2	$87.0 \pm 3.9\%$	19.2 ± 1.7
AQ15	$86.2 \pm 2.6\%$	31.2 ± 2.1

Table 8. Experimental Results: Accuracy and Number of Rules (Meningitis)

Method	Accuracy	No. of Rules
PRIMEROSE-INC	$81.5 \pm 3.2\%$	52.3 ± 1.4
PRIMEROSE	$81.5 \pm 3.2\%$	52.3 ± 1.4
PRIMEROSE0	$72.1 \pm 2.7\%$	12.9 ± 2.1
C4.5	$74.0 \pm 2.1\%$	11.9 ± 3.7
CN2	$75.0 \pm 3.9\%$	33.1 ± 4.1
AQ15	$80.7 \pm 2.7\%$	32.5 ± 2.3

7 Conclusions

Extending concepts of rule induction methods based on rough set theory, we have introduced a new approach to knowledge acquistion, which induces probabilistic rules incrementally, called PRIMEROSE-INC (Probabilistic Rule Induction Method based on Rough Sets for Incremental Learning Methods). This method first uses coverage rather than accuracy, to search for the candidates of rules, and secondly uses accuracy to select from the candidates. When new examples are added, firstly, the method revise the coverage and an accuracy of each elementary attribute value pairs. Then, for each pair, if coverage value decreases, then

it stores it into a removal-candidate list. Else, it stores it into an acceptance-candidate list. Thirdly, for each pair in the removal candidate list, the method searches for a rule including this paer and check whether the accuracy and coverage are larger than given thresholds. Then, the same process is applied to each pair in the acceptance-candidate list. For other rules, the method revises accuracy and coverage.

This system was evaluated on clinical databases on headache and meningitis. The results show that PRIMEROSE-INC induces the same rules as those induced by PRIMEROSE, which extracts rules from all the datasets, but that the former method requires much computational resources than the latter approach.

References

1. Breiman, L., Freidman, J., Olshen, R., Stone, C.: Classification And Regression Trees. Wadsworth International Group, Belmont (1984)
2. Clark, P., Niblett, T.: The cn2 induction algorithm. Machine Learning 3 (1989)
3. Michalski, R.S., Mozetic, I., Hong, J., Lavrac, N.: The multi-purpose incremental learning system aq15 and its testing application to three medical domains. In: AAAI, pp. 1041–1047 (1986)
4. Pawlak, Z.: Rough Sets. Kluwer Academic Publishers, Dordrecht (1991)
5. Quinlan, J.: C4.5 - Programs for Machine Learning. Morgan Kaufmann, Palo Alto (1993)
6. Shan, N., Ziarko, W.: Data-based acqusition and incremental modification of classification rules. Computational Intelligence 11, 357–370 (1995)
7. Skowron, A., Rauszer, C.: The discerniblity matrix and functions in information systems. In: Slowinski, R. (ed.) Intelligent Decision Support. Handbook of Application and Advances of the Rough Set Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
8. Tsumoto, S.: Automated induction of medical expert system rules from clinical databases based on rough set theory. Information Sciences 112, 67–84 (1998)
9. Utgoff, P.E.: Incremental induction of decision trees. Machine Learning 4, 161–186 (1989)
10. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46, 39–59 (1993)

Mining Classification Rules for Detecting Medication Order Changes by Using Characteristic CPOE Subsequences

Hidenao Abe and Shusaku Tsumoto

Department of Medical Informatics, Shimane University, School of Medicine
89-1 Enya-cho, Izumo, Shimane 693-8501, Japan
abe@med.shimane-u.ac.jp, tsumoto@computer.org

Abstract. Computer physician order entry (CPOE) systems play an important role in hospital information systems. However, there are still remaining order corrections and deletions, caused by both of changes of patients' condition and operational problems between a CPOE system and medical doctors. Although medical doctors know a relationship between numbers of order entries and order changes, more concrete descriptions about the order changes are required. In this paper, we present a method for obtaining classification rules of the order changes by using characteristic order entry subsequences that are extracted from daily order entry sequences of patients. By combining patients' basic information, numbers of orders, numbers of order corrections and deletions, and the characteristic order entry subsequences, we obtained classification rules for describing the relationship between the numbers and the order entry changes as a case study. By comparing the contents of the classification rules, we discuss about usefulness of the characteristic order entry sub-sequences for analyzing the order changing factors.

1 Introduction

Recently, computer physician order entry (CPOE) has been introduced as a part of many hospital information systems. The advantage by the computerization of order information is given the improvement of the certainty and promptness, and contributes to doing the communication between medics more smoothly. However, correspondence by the CPOE system that achieves the order entry so that the influence should not go out to a safe medical treatment by correcting and deleting order information (Hereafter, it is called "order change") is needed as described in [3]. The change in these orders should exist together the one thought to be a problem in the one and the system in the treatment process, clarify the patterns in what factor to cause of each, and improve the system.

However, such order entry changes are caused by complex factors including the changes of patient's conditions and the problem between medical doctors and CPOE systems. Although some medical doctors aware the relationship between numbers of order entry and order changes because of difficult conditions of a patient, the implicit knowledge has not been shared among medical stuffs including system engineers. In order to share such knowledge from the view of stored data utilization, the descriptions including patient information, the numbers of order entries, and more concrete features related to the medical processes are needed.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 80–89, 2011.

In this study, we aim to clarify the factor of order changes by using both of the numbers of representative order entries and the input sequence of CPOEs. In order to extract the characteristic partial order sequences, we introduce an automatic term extraction method in natural language processing with regarding each order entry as words. We extract the characteristic partial order sequences identified regardless of the presence of the order change, and propose a method for obtaining classification rules that represent relationships between the partial order sequences and the order change.

In this paper, we describe about the method in Section 2 firstly. Then, by using a order history data from a university hospital, we process datasets consisting of patients' basic information, numbers of order for a day, numbers of order changes, and other information as attributes for each data in Section 3. As for the datasets for a comparison, we obtained two datasets. One includes the characteristic partial order sequences, and the other does not includes them. After obtaining the datasets, we applied if-then rule mining algorithms to obtain rule sets that represent the relationships between the attributes and the medication order changes in Section 4. In this comparison, we compare the accuracies and contents of the rules. Finally, we conclude this in Section 5.

2 A Method to Obtain Classification Rules with Characteristic Order Subsequences

In order to obtain more concrete descriptions that represent the relationships between the characteristic partial order entry sequences and order changes, we combined patients' basic information, the counts of order entries, characteristic order entry subsequences that are ordered in a period.

The process of the method is described as the followings:

- Obtaining datasets for rule mining: extracting basic information of each patient, counting order entry and order changes for a period for each patient, gathering order entry sequences for a period for each patient
- Characteristic partial order sequence extraction: extracting characteristic partial order entry sequences by using a sequential pattern mining method
- Rule generation: applying a classification rule mining algorithm

Based on the numbers of order entries, we can obtain the rules to understand the volume of order entries that causes medication order changes in our previous study. However, the processes of the order entries were not clear in that study, because the counts cannot express the order of the entries. In order to describe the situation of order changes more concretely, we propose to use characteristic order entry sequences included in sequences of CPOEs of each patient.

By assuming the order entries in a period as one sequence, the characteristic order entry subsequences are extracted by using a sequential pattern mining method. Sequential pattern mining is a unsupervised mining approach for extracting meaningful subsequences from the dataset, that consists of ordered items as one instance, by using an index such as frequency [2]. The meaningfulness depends on the application of the sequential pattern mining algorithms. In the following case study, we used an automatic

term extraction method from natural language processing for extracting characteristic order entry subsequences without gaps between two order entries.

Subsequently, by using the appearances of the extracted characteristic order entry subsequences and other features of the order entries in each period, the system extracts the relationships between a target order entry changes and the features. In order to obtain comprehensive descriptions about the relationships for human medical experts, classification rule mining is used to extract the relationships as if-then rules.

3 Processing Order History Data to the Dataset for Rule Mining

In order to obtain the datasets for rule mining, we firstly extracting basic information of each patient form the order history data. Then, we count the numbers of order entries and order changes for a day for each patient. At the same time, we gather order entry sequences for a day for each patient. The process is illustrated as shown in Figure 1. We assume log data of a hospital information system as the original history data of the order entries consisting of ordered patient ID, whose basic information, timestamp of each order, order physician, whose section, order type, and details of each order. According to the timestamps and the patient IDs, the system gathers the following attribute values, sequences of order entries, and class values in a given period for each patient.

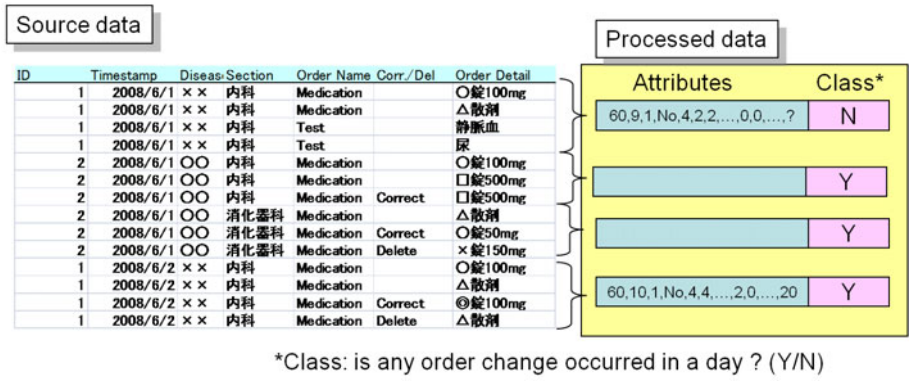


Fig. 1. Overview of the pre-processing of the dataset for rule mining analysis

3.1 Gathering Patient Basic Information

Firstly, we gathered the basic information of patients included in the monthly history data. Since the original patient IDs and the patient names are eliminated previously, we gathered their sex, age, and number of disease by identifying masked ID numbers. The basic features of the patients are appended to the following counts of the daily orders, their changes, the appearances of characteristic order entry subsequences, and the class values. Thus, we set up each day as the period for constructing each instance for rule mining in the following case study.

3.2 Constructing the Instances of the Dataset for Rule Mining

After gathering the patient basic information, the system constructed instances for the rule mining. The instance consists of the patient basic information, the daily features. If a patient consulted two or more sections in a day, two instances are obtained for the results of the consultations in each section. For the abovementioned process, the features related the sections and the consulted hour are also added to the instances.

In order to find out the relationships between a target order changes and other order entries and order changes, we counted the following five representative order entries and their total:

- Total of order entries
- Medication order (Med)
- Laboratory test order (LabTest)
- Physiological function test order (PhyTest)
- Injection order (Inj)
- Rehabilitation order (Reha)

At the same time, we counted the numbers of corrections and deletions of the five order entries.

- Frequencies of order corrections
  - Medication order (MedCrr)
  - Laboratory test order (LabTestCrr)
  - Physiological function test order (PhyTestCrr)
  - Injection order (Inj)
  - Rehabilitation order (RehaCrr)
- Frequencies of order deletions
  - Medication order (MedDel)
  - Laboratory test order (LabTestDel)
  - Physiological function test order (PhyTestDel)
  - Injection order (InjDel)
  - Rehabilitation order (RehaDel)

In addition, we also calculated the difference values between the simple counts of the order entries and their changes. Thus, the 22 features were constructed as the numbers of order entries in a day for each patient in the case study.

3.3 Extracting Characteristic Order Entry Subsequences by Using Sequential Pattern Mining Approach

In the abovementioned daily order entries, the entries make one ordered sequence because each entry has the timestamps. For the order entry sequence corresponded to each instance, we can apply sequential pattern mining for obtaining characteristic partial order entry sequences.

As for the sequential pattern mining approach, there are two major kinds of algorithms developed from different contexts. The algorithms in the one group obtain the

subsequences with gaps such as PrefixSpan [5] and other frequent sequential pattern mining algorithms [2]. The algorithms in the other group obtain the subsequences without gaps such as n-gram [8].

As for considering the relationship between order changes and the concrete order entry subsequences, the subsequences without gaps are more comprehensive than that with gaps. In the following case study, we take a continuing sequential pattern mining algorithm, which was developed for mining more meaningful terms consisting of one or more words automatically. Then, the appearances(0/1) of the top N characteristic order entry subsequences for each instance are added to the datasets.

4 A Case Study on an Actual Order Entry Histories

In this section, we used an actual computer physician order entry data that were obtained in a Japanese hospital that has more than 600 beds and 1000 outpatients a day. By applying a sub-sequence extraction method, we extracted characteristic partial order entry sequences related to medication order changes. The relationships are described by if-then rules in the followings.

4.1 Counts of the Five Kinds of Order Entries and the Medication Order Entry Changes

According to the pre-processing process as described in Section 3, we obtained datasets for rule mining algorithms that are consisting of basic information of the patients, counts of the orders, and the appearances of extracted characteristic order entry subsequences.

We counted the two monthly order entry history data for June 2008 and June 2009. Since most Japanese hospital for outpatients is timeless to input the medical records, we separated the processed dataset for outpatients and inpatients. Considering the difference of the time limitation of each consultation, we counted the numbers of the order entries and their changes separately. The statistics of the numbers of the 22 order entries as shown in Table 1. Excepting the number of the medication order and the total number of the daily order entries, the minimum number of these order entries is 0.

Based on the numbers of order entries, we set up the class for the medication order entry changes by using the corrections of medication order and the deletions. The class distributions of the datasets for the two months, June 2008 and June 2009, are shown in Table 2.

In order to avoid tautological rules from the datasets, we removed the features directly counting the class for the following analysis; MedCrr and MedDel.

4.2 Extracting Characteristic Order Entry Subsequences by Using Term Extraction Method in Natural Language Processing

In order to constructs the features of the characteristic order entry subsequences, we performed the following steps:

Table 1. The maximum and averaged numbers of daily counts of the five order entries

	June 2008				June 2009			
	Inpatient		Outpatient		Inpatient		Outpatient	
Att.	max	avg.	max	avg.	max	avg.	max	avg.
Total	47	4.24	47	3.89	58	4.32	63	3.94
Med	44	2.54	47	3.05	48	2.65	62	3.09
LabTest	19	0.56	11	0.54	19	0.50	10	0.54
PhyTest	8	0.07	8	0.11	7	0.08	8	0.10
Inj	29	1.00	22	0.15	39	0.96	17	0.17
Reha	5	0.09	14	0.04	14	0.11	9	0.04
MedCrr	25	0.20	21	0.08	32	0.22	13	0.09
LabTestCrr	13	0.08	7	0.02	10	0.05	5	0.01
PhyTestCrr	3	0.01	2	0.01	4	0.01	1	0.01
InjCrr	8	0.09	7	0.03	10	0.10	12	0.02
RehaCrr	2	0.00	12	0.00	3	0.00	3	0.00
MedDel	20	0.06	20	0.02	14	0.05	27	0.01
LabTestDel	3	0.02	3	0.01	4	0.02	2	0.02
PhyTestDel	2	0.00	2	0.00	2	0.00	2	0.00
InjDel	5	0.01	5	0.00	21	0.01	4	0.00
RehaDel	3	0.00	2	0.00	6	0.00	3	0.00
TotalSub	31	3.75	30	3.72	37	3.84	36	3.77
MedSub	29	2.28	29	2.95	36	2.39	35	2.99
LabTestSub	13	0.47	11	0.51	13	0.42	8	0.51
PhyTestSub	7	0.06	8	0.10	7	0.06	6	0.09
InjSub	21	0.85	15	0.13	27	0.85	13	0.14
RehaSub	5	0.09	5	0.03	8	0.10	6	0.04

Table 2. Class distributions about the medication order changes on June 2008 and June 2009

		Overall	Without Med. Order Changes	With Med. Order Changes
June 2008	Inpatient	7136	6607	529
	Outpatient	10201	9863	338
June 2009	Inpatient	7426	6864	56
	Outpatient	10509	10096	413

1. Gathering daily order entry sequences of the patient
2. Applying a sequential pattern mining algorithm for extracting characteristic order entry subsequences from the overall sequences
3. Selecting top N characteristic order entry subsequences under the unsupervised manner
4. Obtaining the features corresponding to the characteristic order entry subsequences to the dataset for the rule mining

In this case study, we applied an automatic term extraction method [4] for extracting characteristic partial order sequences in the set of order entry sequences by assuming the order entry sequences as documents.

This method involves the detection of technical terms by using the following values for each candidate CN:

$$FLR(CN) = f(CN) \times (\prod_{i=1}^{L}(FL(N_i)+1)(FR(N_i)+1))^{\frac{1}{2L}}$$

Table 3. Top 10 characteristic partial order sequences based on FLR scores: (a) on June 2008, (b) on June 2009

(a) June 2008

Inpatient		Outpatient	
Order entry subsequence	FLR score	Order entry subsequence	FLR score
Med	12.17	Med	11.45
Med Med	9.66	Med Med	10.48
HospCtrl Med	9.00	PhyTest med	10.11
Med HospCtrl	8.94	Med PhyTest Med	9.72
Med PhyTest	8.74	PhyTest Med Med	9.37
PhyTest Med	8.74	Med PhyTest	9.33
Med Med Med	8.40	Med Med Med Med	8.93
Med Reha	8.13	PhyTest Med Med Med	8.53
HospCtrl Med Med	7.88	Med Med Med Med Med	8.41
Reha Med	7.72	HospCtrl Med	8.26

(b) June 2009

Inpatient		Outpatient	
Order entry subsequence	FLR score	Order entry subsequence	FLR score
Med	12.14	Med	11.54
Med Med	9.56	Med Med	10.52
HospCtrl Med	8.97	PhyTest Med	10.09
Med HospCtrl	8.78	Med Med Med	9.71
Med LabTest	8.62	Med PhyTest	9.44
Med Med Med	8.55	PhyTest Med Med	9.35
PhyTest Med	8.55	Med Med Med Med	9.04
Med Reha	8.36	Med Med Med Med Med	8.65
Reha Med	8.17	PhyTest Med Med Med	8.65
HospCtrl Med Med	8.09	HospCtrl Med	8.31

where $f(CN)$ means frequency of a candidate CN appeared isolated, and $FL(N_i)$ and $FR(N_i)$ indicate the frequencies of different orders on the right and the left of each order entry N_i in bi-grams including each CN. In the experiments, we selected technical terms with this FLR score as $FLR(t) > 1.0$.

By applying this method to the two dataset on June 2008 and June 2009, the system extracted partial order sequences respectively. The top ten characteristic partial order sequences are shown in Table 3. Since these partial order sequences are determined whole of order entry sequences, we do not know about which ones are related to the target order changes in this stage. However, these partial order sequences are not only appeared frequently but also used some characteristic situations.

For example, "Med LabTest" means continued one medication order entry and one laboratory test order entry. This partial order sequence is more characteristic in the outpatient situations than "Med Med Med" (three continued medication order entries) based on their FLR scores. Although there are the differences of the extracted subsequences between the inpatient sequences and outpatient sequences, the top 10 characteristic subsequences are the same on both of the months from different years.

Table 4. Averaged accuracies(%) and their SDs of classifiers on each dataset

2008 June	Inpatient				Outpatient			
	Without Subsequences		With Subsequences		Without Subsequences		With Subsequences	
	Avg. Acc(%)	SD	Avg. Acc(%)	SD	Avg. Acc(%)	SD	Avg. Acc(%)	SD
NaiveBayes	87.36	1.12	86.80	1.15	90.75	0.82	89.59	0.94
k-NN	90.46	0.89	89.53	0.99	94.34	0.63	94.70	0.60
OneR	***95.23***	0.48	***95.23***	0.48	96.70	0.48	96.70	0.48
C4.5(J48)	***99.14***	0.36	***99.13***	0.36	***98.54***	0.40	***98.54***	0.39
PART	***99.40***	0.39	***99.34***	0.43	***98.49***	0.40	***98.49***	0.41
% Majority	92.60		92.60		96.20		96.20	

2009 June	Inpatient				Outpatient			
	Without Subsequences		With Subsequences		Without Subsequences		With Subsequences	
	Avg. Acc(%)	SD	Avg. Acc(%)	SD	Avg. Acc(%)	SD	Avg. Acc(%)	SD
NaiveBayes	85.86	1.20	86.35	1.17	90.41	0.87	88.84	1.02
k-NN	89.57	1.01	87.73	1.16	93.73	0.82	95.25	0.51
OneR	***95.34***	0.51	***95.34***	0.51	***97.28***	0.45	***97.28***	0.45
C4.5(J48)	***99.25***	0.34	***99.28***	0.33	***99.48***	0.31	***99.47***	0.33
PART	***99.46***	0.33	***99.48***	0.32	***99.22***	0.45	***99.24***	0.35
% Majority	92.40		92.40		96.10		96.10	

After extracting the top 10 characteristic order entry subsequences, we obtained the features corresponding to the subsequences as their appearances. If a characteristic order entry subsequence appears in the daily sequence of order entries, the value of the feature is '1'. Otherwise, the value is '0'. This process is repeated to the top 10 characteristic order entry subsequences for each daily order entry sequence of the patient.

4.3 Obtaining Classification Rules with/without Characteristic Order Entry Subsequences

In this section, we compare the accuracies on the datasets including the appearances of top 10 characteristic order entry subsequences to without them by using five representative classification learning algorithms. Then, we discuss the contents of the two rule sets. We used the four datasets the numbers of order entries with/without the appearances of the top 10 characteristic order entry subsequences on the different two months and the place of the entries.

In order to evaluate the effect of the characteristic order entry sequences for the classification of the medication order changes, we performed 100 times repeated 10-fold cross validation on the following five classification learning algorithms; NaiveBayes, k-NN($k = 5$), C4.5 [6], OneR [7], and PART [1]. We used their implementations on Weka [9].

As shown in Table 4, the emphasized averaged accuracies in Table 4 significantly outperform the percentages of the majority class label; without medication order changes. The significance of the difference is tested by using one-tail t-test with $\alpha = 0.05$ as the significance level. As shown in this result, C4.5 and PART obtain more accurate classification models by using both of the feature sets; with/without the appearances of the top 10 characteristic order entry subsequences. The result for the June 2009 datasets indicates that the characteristic partial order entry sequences mean the relationships between the order entry sequences and the medication order changes.

In order to compare the availability of the partial order entry sequences for detecting medication order changes, we observed the contents of the representative rules obtained

for each dataset. The classifiers of PART achieve the highest accuracies to the dataset with generating classification rules based on the information gain ratio by separating a given dataset into subsets. The representative rules with/without the partial order sequences are shown in Figure 2.

The rules without the characteristic order entry subsequences cover large number of instances for each dataset. However, these rules do not express any order of the entries, and the rule obtained from the datasets of June 2009 for inpatients is very difficult to understand. So, medical doctors pointed out that it is difficult to image concrete medical situations from the rules.

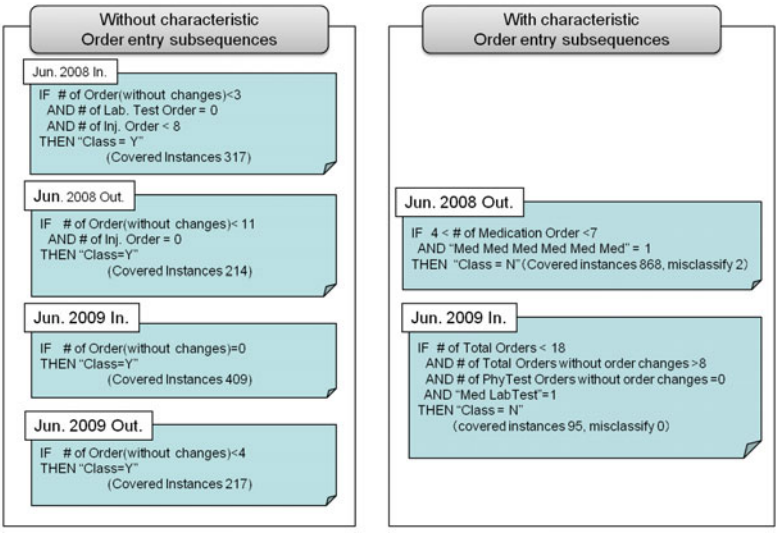


Fig. 2. The representative rules with highest coverage for each entire training dataset

On the other hand, the rules containing the characteristic order partial sequences are shown in Figure 2. Although these rules represent the relationships between the partial order entry sequences and the result without order changes, the coverage and the correctness of these rules are high. Although the rules with the characteristic order entry subsequences cover only few instances, their accuracies are higher than the rules that are constructed with the numbers of order entries. Based on this result, the order partial sequences distinguish the result without the medication order changes more concretely.

5 Conclusion

In this paper, we present a rule mining process for detecting CPOE changes by using numbers of order entries and characteristic partial physician order entry sequences on the log data. In order to extract the characteristic order entry subsequences, we introduce the automatic term extraction method that has been developed as a natural language processing method.

By applying the classification rule mining, we obtained more accurate classification rules that outperform the percentages of the majority class labels. Based on the result, the characteristic partial order entry sequences can distinguish the situations without any medication order change for a day more clearly.

In the future, we will combine this method with more detailed medication order subsequences such as sequences of the names of drugs. We will also analyze the temporal patterns for the factors related to the order changes based on the attributes of this analysis and the textual descriptions on the medical documents.

References

1. Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I.H.: Using model trees for classification. Machine Learning 32(1), 63–76 (1998)
2. Mabroukeh, N.R., Ezeife, C.I.: A taxonomy of sequential pattern mining algorithms. ACM Comput. Surv. 43, 3:1–3:41 (2010), http://doi.acm.org/10.1145/1824795.1824798
3. Magrabi, F., McDonnell, G., Westbrook, J., Coiera, E.: Using an accident model to design safe electronic medication management systems. Stud. Health Technol. Inform. 129, 948–952 (2007)
4. Nakagawa, H.: Automatic term recognition based on statistics of compound nouns. Terminology 6(2), 195–210 (2000)
5. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: Proc. of the 17th International Conference on Data Engineering, pp. 215–224. IEEE Computer Society, Los Alamitos (2001)
6. Quinlan, J.R.: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
7. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Machine Learning 11, 63–91 (1993)
8. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423,623–656 (1948)
9. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (2000)

Incorporating Neighborhood Effects in Customer Relationship Management Models

Philippe Baecke and Dirk Van den Poel

Ghent University, Faculty of Economics and Business Administration,
Department of Marketing, Tweekerkenstraat 2, B-9000 Ghent, Belgium

Abstract. Traditional customer relationship management (CRM) models often ignore the correlation that could exist in the purchasing behavior of neighboring customers. Instead of treating this correlation as nuisance in the error term, a generalized linear autologistic regression can be used to take these neighborhood effects into account and improve the predictive performance of a customer identification model for a Japanese automobile brand. In addition, this study shows that the level on which neighborhoods are composed has an important influence on the extra value that results from the incorporation of spatial autocorrelation.

Keywords: Customer Intelligence; Predictive Analytics; Marketing; Data Augmentation; Autoregressive Model; Automobile Industry.

1 Introduction

Besides the data mining technique, the success of a CRM model also depends on the quality of the information used as input for the model [1]. Traditional CRM models often ignore neighborhood information and rely on the assumption of independent observations. This means that customers' purchasing behavior is totally unrelated to the behavior of others. However, in reality, customer preferences do not only depend on their own characteristics but are often related to the behaviors of other customers in their neighborhood. Using neighborhood information to incorporate spatial autocorrelation in the model can solve this violation and significantly improve the predictive performance of the model.

Several studies have already proven that spatial statistics can produce interesting insights in marketing [2-8]. However, only a limited number of studies use spatial information to improve the accuracy of a predictive CRM model. In reference [9], customer interdependence is estimated based on geographic and demographic proximity. The study indicates that geographic reference groups are more important than demographic reference groups in determining individual automobile preferences. Reference [10] shows that taking zip-code information into account can significantly improve a model used for the attraction of new students by a private university. The focus of this study will also be on incorporating physical geographic interdependence to improve CRM models, but, compared to this previous literature, this study includes a large number of independent socio-demographic and lifestyle variables that are typically available at an external data vendor. This should avoid that the predictive improvement could

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 90–95, 2011.

be caused by the absence of other important variables that can easily be obtained for customer acquisition models.

In addition, this article introduces an extra complexity that is mostly ignored in previous literature. Customers can often be clustered in neighborhoods at multiple levels (e.g. country, district, ward, etc.). In order to incorporate these neighborhood effects efficiently, the level of granularity should be carefully chosen. If the neighborhood is chosen too large, interdependences will fade away because the preferences of too many surrounding customers are taken into account that do not have any influence in reality. On the other hand, choosing neighborhoods that are too small can affect the reliability of the measured influence and ignore the correlation with some customers that still have an influence. This study will compare the relevance of taking spatial neighborhood effects into account at different levels of granularity.

In this paper, neighborhood information is used to incorporate spatial autocorrelation in a customer acquisition model for a Japanese car brand. Within CRM models, customer acquisition models suffer often the most from a lack of data quality. A company's customer database is typically single source in nature. The data collection is limited to the information a company retrieves from its own customers. As a result, for customer acquisition campaigns the company has to attract data from external data vendors. Nevertheless, this data still only contains socio-demographic and lifestyle variables [11]. Especially in such situation, incorporating extra neighborhood information can improve the identification of potential customers.

The remainder of this article is organized as follows. Section 2 describes the methodology, consisting of the data description and the generalized linear autologistic regression model used in this study. Next, the results are reported in Section 3 and this paper ends with conclusions in Section 4.

2 Methodology

2.1 Data Description

Data is collected from one of the largest external data vendors in Belgium. This external data vendor possesses data about socio-demographics and lifestyle variables from more than 3 million respondents in Belgium. Furthermore, it provides information about automobile ownership in December 2007 of a Japanese automobile brand.

The model in this study has a binary dependent variable, indicating whether the respondent owns the Japanese automobile brand. Based on this model, potential customers with a similar profile as the current owners can be identified. These prospects can then be used in a marketing acquisition campaign. Because a customer acquisition model typically cannot rely on transactional information, 52 socio-demographic and lifestyle variables are included as predictors.

Further, also information about the geographical location of the respondents is available. Table 1 illustrates that respondents can be divided into several mutually exclusive neighborhoods at different levels of granularity. This table presents seven granularity levels together with information about the number of neighborhoods at each level, the average number of respondents and the average number of owners in each neighborhood.

Table 1. Overview of the granularity levels

Granularity level	Number of neighborhoods	Average number of respondents	Average number of owners
level 1	9	349281.78	3073.00
level 2	43	73105.49	643.19
level 3	589	5337.07	46.96
level 4	3092	1016.67	8.94
level 5	6738	466.54	4.10
level 6	19272	163.11	1.44
level 7	156089	20.14	0.18

Analysis based on a finer level of granularity will divide the respondents over more neighborhoods resulting in a smaller number of interdependent neighbors. At the finest level, an average of about 20 respondents is present in each neighborhood, which corresponds with an average of only 0.18 owners per neighborhood. This study will investigate which granularity level is optimal to incorporate customer interdependence using a generalized linear autologistic regression model.

2.2 Generalized Linear Autologistic Regression Model

A typical data mining technique used in CRM to solve a binary classification problem is a logistic regression. This model is very popular in CRM because of its interpretability. Unlike other, more complex predictive techniques (e.g. neural networks), logistic regression is able to provide information about the size and direction of the effects of independent variables [12,13].

A key assumption of this traditional model is that the behavior of one individual is independent of the behavior of another individual. Though, in reality, a customers' behavior is not only dependent of its own characteristics but is also influenced by the preferences of others. In traditional data mining techniques this interdependence is treated as nuisance in the error term. However, an autologistic regression model can be used to consider spatial autocorrelation explicitly in a predictive model for a binary variable [6,14-16]. The generalized linear autologistic regression model in this study is a modified version of the general autologistic model used in reference [17]:

$$\mathrm{P}(\mathrm{y} = 1 \,|\text{all other values}) = \frac{\exp(\eta)}{1 + \exp(\eta)}. \tag{1}$$

$$\text{Where } \eta = \beta_0 + \mathrm{X}\beta_1 + \rho WY .$$

In this equation a logit link function is used to adopt the regression equation to a binomial outcome variable. Whereby Y is an n x 1 vector of the dependent variable; X is an n x k matrix containing the explanatory variables; the intercept is represented by β_0.and β_1 is a k x 1 vector of regression coefficients to be estimated. This model includes a spatial lag effect by means of the autoregressive coefficient ρ to be estimated for the spatially lagged dependent variables WY.

These spatially lagged dependent variables are constructed based on a spatial weight matrix W. This is an n x n matrix containing non-zero elements w_{ij} indicating

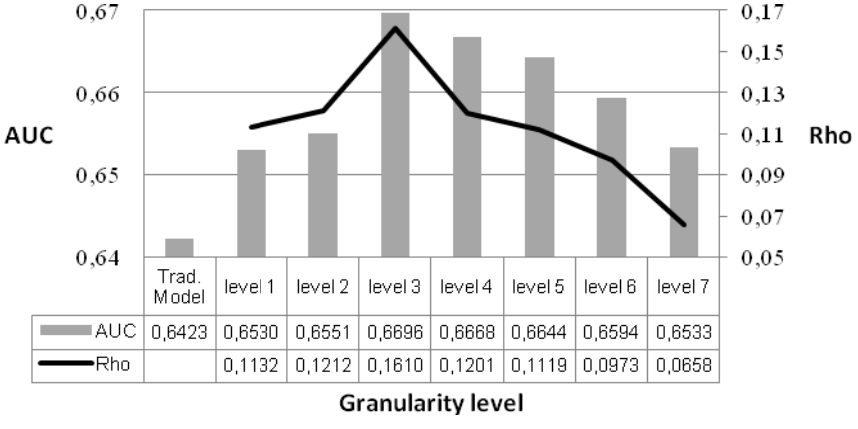
interdependence between observation i (row) and j (column). By convention, self-influence is excluded such that diagonal elements w_{ii} equal zero. Next, this weight matrix is row-standardized using the following formula:

$$w_{ij}^{s} = \frac{w_{ij}}{\Sigma_j w_{ij}}. \tag{2}$$

The weight matrix is an important element in a generalized linear autologistic regression model. This study investigates how the choice of neighborhood level can influence the predictive performance of the customer acquisition model. For customers living in the same neighborhood w_{ij} will be set to one in the non-standardized weight matrix. Hence, at a coarse granularity level the number of neighborhoods is small resulting in a high number of interdependent relationships included in the weight matrix. As the granularity level becomes finer, the number of non-zero elements in the weight matrix will drop.

3 Results

In Figure 1, the traditional customer identification model and all spatial models at different levels of granularity are compared. This figure presents for each model the predictive performance on the validation sample in terms of AUC [18] and the autoregressive coefficients estimated by the spatial models.



	Trad. Model	level 1	level 2	level 3	level 4	level 5	level 6	level 7
AUC	0,6423	0,6530	0,6551	0,6696	0,6668	0,6644	0,6594	0,6533
Rho		0,1132	0,1212	0,1610	0,1201	0,1119	0,0973	0,0658

Fig. 1. Overview of the AUCs and the spatial autoregressive coefficients

This spatial autoregressive coefficient is positive and significantly different from zero in all autologistic regressions. This suggests the existence of interdependence at all granularity levels. In other words, the average correlation between automobile preferences of respondents in the same neighborhood is higher than the average correlation between automobile preferences of respondents located in different neighborhoods. Comparing the AUC indicators of the spatial models with the benchmark traditional logistic regression model, using the non-parametric test of Delong et al.

[19], demonstrates that incorporating these neighborhood effects significantly improves the accuracy of the acquisition model.

Figure 1 illustrates that the proportion of this predictive improvement heavily depends on the chosen granularity level. The optimal predictive performance in this study is achieved at granularity level 3. If the neighborhood level is too coarse, correlation is assumed between too many customers that do not influence each other in reality. On the other hand, a granularity level that is too fine can affect the reliably of the measured autocorrelation and ignore interdependences that exist in reality. A similar evolution can be found in the spatial autoregressive coefficient, which represents the existence of spatial interdependence in the model.

Comparing the predictive performance of a customer acquisition model that incorporates neighborhood effects at the optimal granularity level with the benchmark traditional logistic regression model illustrates that taking spatial correlation into account increases the AUC by 0.0273. This is not only statistically significant, but also economically relevant and should help the marketing decision maker to improve his customer acquisition strategies.

4 Conclusions

Traditional customer acquisition models often ignore the spatial correlation that could exist between the purchasing behaviors of neighboring customers and treats this as nuisance in the error term. This study shows that, even in a model that already includes a large number of socio-demographic and lifestyle variables typically attracted for customer acquisition, Extra predictive value can still be obtained by taking this spatial interdependence into account using a generalized linear autologistic regression model.

Moreover, this study illustrates that the choice of neighborhood level in such an autologistic model has an important impact on the model's accuracy. Using a granularity level that is too coarse or too fine respectively incorporates too much or too little interdependence in the weight matrix resulting in a less than optimal predictive improvement.

Acknowledgement

Both authors acknowledge the IAP research network grant nr. P6/03 of the Belgian government (Belgian Science Policy).

References

1. Baecke, P., Van den Poel, D.: Improving Purchasing Behavior Predictions by Data Augmentation with Situational Variables. Int. J. Inf. Technol. Decis. Mak. 9, 853–872 (2010)
2. Bradlow, E.T., Bronnenberg, B., Russell, G.J., Arora, N., Bell, D.R., Duvvuri, S.D., Ter-Hofstede, F., Sismeiro, C., Thomadsen, R., Yang, S.: Spatial Models in Marketing. Mark. Lett. 16, 267–278 (2005)
3. Bronnenberg, B.J.: Spatial models in marketing research and practice. Appl. Stoch. Models. Bus. Ind. 21, 335–343 (2005)

4. Bronnenberg, B.J., Mahajan, V.: Unobserved Retailer Behavior in Multimarket Data: Joint Spatial Dependence in Market Shares and Promotional Variables. Mark. Sci. 20, 284–299 (2001)
5. Bell, D.R., Song, S.: Neighborhood effects and trail on the Internet: Evidence from online grocery retailing. QME-Quant. Mark. Econ. 5, 361–400 (2007)
6. Moon, S., Russel, G.J.: Predicting Product Purchase from Inferred Customer Similarity: An Autologistic Model Approach. Mark. Sci. 54, 71–82 (2008)
7. Grinblatt, M., Keloharju, M., Ikäheimo, S.: Social Influence and Consumption: Evidence from the Automobile Purchases of Neighbors. Rev. Econ. Stat. 90, 735–753 (2008)
8. Manchanda, P., Xie, Y., Youn, N.: The Role of Targeted Communication and Contagion in Product Adoption. Mark. Sci. 27, 961–976 (2008)
9. Yang, S., Allenby, G.M.: Modeling Interdependent Customer Preferences. J. Mark. Res. 40, 282–294 (2003)
10. Steenburgh, T.J., Ainslie, A.: Massively Categorical Variables: Revealing the Information in Zip Codes. Mark. Sci. 22, 40–57 (2003)
11. Baecke, P., Van den Poel, D.: Data augmentation by predicting spending pleasure using commercially available external data. J. Intell. Inf. Syst., doi:10.1007/s10844-009-0111-x
12. McCullagh, P., Nelder, J.A.: Generalized linear models. Chapman & Hall, London (1989)
13. Hosmer, D.W., Lemeshow, S.: Applied Logistic Regression. John Wiley & Sons, New York (2000)
14. Augustin, N.H., Mugglestone, M.A., Buckland, S.T.: An Autologistic Model for the Spatial Distribution of wildlife. J. Appl. Ecol. 33, 339–347 (1996)
15. Hoeting, J.A., Leecaster, M., Bowden, D.: An Improved Model for Spatially Correlated Binary Responses. J. Agric. Biol. Environ. Stat. 5, 102–114 (2000)
16. He, F., Zhou, J., Zhu, H.: Autologistic Regression Model for the Distribution of Vegetation. J. Agric. Biol. Environ. Stat. 8, 205–222 (2003)
17. Besang, J.: Spatial Interaction and the Statistical Analysis of Lattice Systems. J. Roy. Statist. Soc. Ser. B (Methodological) 36, 192–236 (1974)
18. DeLong, E.R., DeLong, D.M., Clarke-Pearson, D.L.: Comparing the areas under two or morecorrelated receiver operating characteristic curves: a nonparametric approach. Biometrics 44, 837–845 (1988)
19. Hanley, J.H., McNeil, B.J.: The meaning and use of area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36 (1982)

ETree Miner: A New GUHA Procedure for Building Exploration Trees

Petr Berka[1,2]

[1] University of Economics, W. Churchill Sq. 4, 130 67 Prague
[2] Institute of Finance and Administration, Estonska 500, 101 00 Prague

Abstract. Induction of decision trees belongs to the most popular algorithms used in machine learning and data mining. This process will result in a single tree that can be use both for classification of new examples and for description the partitioning of the training set. In the paper we propose an alternative approach that is related to the idea of finding all interesting relations (usually association rules, but in our case all interesting trees) in given data. When building the so called exploration trees, we consider not a single best attribute for branching but more "good" attributes for each split. The proposed method will be compared with the "standard" C4.5 algorithm on several data sets from the loan application domain.

We propose this algorithm in the framework of the GUHA method, a genuine exploratory analysis method that aims at finding all patterns, that are true in the analyzed data.

1 Introduction

GUHA is an original Czech method of exploratory data analysis developed since 1960s. Its principle is to offer all interesting facts following from the given data to the given problem. A milestone in the GUHA method development was the monograph [7], which introduces the general theory of mechanized hypothesis formation based on mathematical logic and statistics. The main idea of the GUHA method is to find all patterns (relations, rules), that are true in the analyzed data.

Recently, various GUHA procedures, that mine for different types of rule-like patterns have been proposed and implemented in the LISp-Miner [12], [13] and Ferda systems [11]. The patterns are various types of relations between pairs of boolean or categorial attributes. Let us consider as an example the 4FT-Miner procedure. This procedure mines for patterns in the form $\varphi \approx \psi/\xi$, where φ (antecedent), ψ (succedent) and ξ (condition) are conjunctions of literals and $\approx$ denotes a relation between φ and ψ for the examples from the analyzed data table, that fulfill ξ (ξ need not to be defined). *Literal* is in tho form $A(\alpha)$ or its negation $\neg A(\alpha)$, where A is an attribute and α) is a subset of its possible values. A 4FT pattern is true in the analyzed data table, if the condition associated with $\approx$ is satisfied for the frequencies a, b, c, d of the corresponding contingency table. Since $\approx$ can be defined e.g. as $\frac{a}{a+b} \geq 0.9$, one type of patterns that can

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 96–101, 2011.

be looked for using 4FT-Miner are association rules as defined by Agrawal e.g. in [1].

The tree building algorithm proposed in this paper fits into the GUHA framework as a procedure that looks for tree-like patterns each defining a particular partition of the analysed data table w.r.t. the given (class) attribute.

2 Induction of Exploration Trees

The TDIDT (top-down induction of decision trees) family of algorithms recursively partitions the attribute space to build rectangular regions that contain examples of one class. This method, also known as "divide and conquer" has been implemented in a number of algorithms like ID3 [9], C4.5 algorithm [10], CART [4] or CHAID [2]. All these algorithms are based on greedy top-down search in the space of possible trees. Once a splitting attribute is chosen, the algorithm proceeds further and no backtracking (and changing the splitting attribute) is possible. The result is thus a single tree.

The decision trees are usually used for classification. In this case, a new example is propagated down the tree at each splitting node selecting the branch that corresponds to the value of the splitting attribute. The leaf of the tree shows then the class for this example. But simultaneously, a decion tree can be interpreted as a pertition of the given data set into subsets, that are homogeneous w.r.t. class attribute.

2.1 The Algorithm

Our proposed algorithm for exploration trees is based on an extension of this approach. Instead of selecting single best attribute to make a split of the data, we select more suitable attributes. The result thus will be a set of trees (forest) where each tree represents one model applicable for both description and classification. The algorithm thus differs from the "standard" TDIDT algorithms in two aspects (see Fig. 1 for a simplified description):

- we use more attributes to make a split (see point 1 of the algorithm),
- we use different stopping criteria to terminate the tree growth (see point 1.2 of the algorithm).

The basic difference is the possibility to use of more attributes to make a split. To decide about the suitability of an attribute to make a split, we use the χ^2 criterion defined as

$$\chi^2 = \sum_i \sum_j \frac{(a_{ij} - \frac{r_i \cdot s_j}{n})^2}{\frac{r_i \cdot s_j}{n}},$$

where a_{ij} is the number of examples that have the i-th value of the attribute A and the j-th value of the target attribute, r_i is the number of examples that have the i-th value of the attribute A, s_j is the number of examples that have the j-th

ETree algorithm

1. for every attribute suitable as a root of the current (sub)tree,
   1.1 divide data in this node into subsets according to the values of the selected attribute and add new node for each this subset,
   1.2 if there is an added node, for which the data do not belong to the same class, goto step 1.

Fig. 1. The ETree algorithm

value of the target attribute and n is the number of all examples. This criterion not only ranks the attributes but also evaluates the "strengths" of the relation between the evaluated and class attributes. So we can consider only significant splitting attributes. The significance testing allows also to reduce the number of generated trees; if the best splitting attribute is not significantly related with the class, we can immediately stop growing tree at this node and need not to consider other splitting possibilities. We can of course use only the best splitting attribute, in this case we get the classical TDIDT algorithm.

We consider several stopping criteria to terminate the tree growing. Beside the standard node impurity (the fraction of examples of the majority class) we can use also the node frequency (number of examples in a node), the depth of the tree and the above mentioned χ^2 test of significance.

The quality of the exploration tree is evaluated using the classification accuracy on training data. This quantity, like the confidence of an association rule, shows how good the tree corresponds to the given data (and only trees that satisfy the given lower bound are produced at the output of the algorithm). So we are primary interested in description not classification; in this situation, each tree should be inspected and interpreted by the domain expert. Nevertheless, the exploration trees can be used for classification as well. Here we can employ different classification strategies (not yet fully implemented):

- classical decision tree: only the best attribute is using for branching, stopping criterion is the node impurity,
- ensemble of best trees: k best attributes used for branching, stopping criterion can be node impurity, node frequency or tree depth, classification based on voting of each of the trees,
- ensemble of decision stumps: all attributes used for branching, the depth of the tree set to 0, classification based on voting of each of the trees.

The algorithm for finding exploration trees is implemented in the Ferda system [11]. This system offers a user friendly visual (graphical) environment to compose and run the data mining tasks realized as various GUHA procedures. The incorporation of ETree algorithm into the LISp-Miner system is under development.

2.2 Empirical Evaluation

The goal of the experiments was to test the ETree algorithm when building description trees. We thus evaluate the results only on the training data. We compared the behavior of our algorithm with the C4.5 algorithm implemented in Weka system [15].

We used several data sets from the UCI Machine Learning Repository [14]. The characteristics of these data are shown in Tab. 1.

We run two types of experiments. Our aim in experiment 1 was to create single, most accurate tree. Tree growing in ETree was thus not restricted by any of the parameters min. node impurity or min. node frequency. In C4.5 we disable pruning of the tree. The results (left part of Table 2) show, that the results of both algorithms are (for 10 out of 11 data sets) the same in terms of classification accuracy on training data. The goal in experiment 2 was to build more exploration trees. In C4.5 we set the minimal number of examples in a leaf to 3. We used the same setting for this parameter in ETree and we further set the max. number of best attributes considered for splitting to 2 and the depth of the tree corresponding to the depth of the tree generated in C4.5 (to have same settings for both algorithms). The results for this experiment (right part of Table 2) show, that ETree founds (for 10 out of 11 data sets) a tree that was better (on the training data) than that generated by C4.5. The number in bold emphasise (for both experiments) the better results.

Table 1. Description of used data

Data	no. examples	no. attributes
Australian credit	690	15
Brest cancer	286	10
Iris	150	5
Japan Credit	125	11
Lenses	24	5
Monk1	123	7
Mushroom	8124	23
Pima indian diabetes	768	9
Tic-tac-toe	958	10
Tumor	339	18
Vote	435	17

3 Related Work

The exploration trees, especially when used for classification, can be understood as a kind of ensembles. From this point of view, we can find some related work in the area of combining tree-based classifiers: AdaBoost creates a sequence of models (trees) where every model in the sequence focuses on examples wrongly classified by its predecessors [6]. Random forrest algorithm builds a set of trees

Table 2. Summary of the results

Data	Experiment 1		Experiment 2	
	C4.5	ETree	C4.5	ETree
Australian credit	0.9913	0.9913	0.9275	**0.9565**
Breast cancer	0.9792	0.9792	0.8671	**0.9214**
Iris	0.9801	0.9801	0.9733	**0.9735**
Japan Credit	1.0000	1.0000	0.8560	**0.9360**
Lenses	1.0000	1.0000	0.9167	0.9167
Monk1	**0.9837**	0.8780	**0.8699**	0.8374
Mushroom	1.0000	1.0000	1.0000	1.0000
Pima indian diabetes	0.9518	0.9518	0.8568	**0.9010**
Tic-tac-toe	1.0000	1.0000	0.9436	**0.9761**
Tumor	0.8289	**0.9357**	0.6017	**0.7535**
Vote	0.9908	0.9908	0.9678	**0.9770**

by randomly splitting the data into training subsets and by randomly selecting a subset of input attributes to build an individual tree [3]. The Option Trees algorithm creates a single tree that beside "regular" branching nodes contains also so called option nodes that include more attributes proposed to make a split. This tree thus represents a set of trees that differ in the splitting attribute used in the option node [8]. The ADTree (alternating decision trees) algorithm builds a sequence of trees with increasing depth. These trees are represented in a compact form of the tree with the maximal depth where each branching node is replaced by a pair [classifying node, branching node], where classifying node is a leaf node at the position of branching [5]. All these algorithms focus on building ensemble classifiers, so the individual trees are not presented to the user.

4 Conclusions

The paper presents ETree, a novel algorithm for building so called exploration trees. It's main difference to standard TDIDT algorithms is the possibility to use more attributes to create a split and thus the possibility to build more trees that describe given data. The resulting trees can be used for both classification and description (segmentation).

We empirically evaluated our algorithm by comparing its classification accuracy (computed for the training data) on some benchmark data with the state-of-the-art algorithm C4.5. The first experiment shows that when giving no restrictions that will reduce the tree growing, the "best" tree generated by both algorithms is usually the same. The second experiment shows, that when building in ETRee Miner more trees with initial setting that corresponds to a pruned version of the tree generated by C4.5, we can usually find (among the trees generated) a better tree than that of C4.5. The reason can be twofold: (1) C4.5 is optimized to perform well on unseen data and thus does not cover the training data as precise as possible, and (2) the greedy search need not to find

the best tree (especially if the selection of the splitting attribute is based on few examples, what is typical for branching nodes far from the root).

Acknowledgement

The work is supported by the grant MSMT 1M06014 (from the Ministry of Education of the Czech Republic) and the grant GACR 201/08/0802 (from the Grant Agency of the Czech Republic).

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: SIGMOD Conference, pp. 207–216 (1993)
2. Biggs, D., deVille, B., Suen, E.: A method of choosing multiway partitions for classification and decision trees. Journal of Applied Statistics 18(1), 49–62 (1991)
3. Breiman, L.: Random Forrests. Machine Learning 45, 5–32 (2001)
4. Breiman, L., Friedman, J.H., Ohlsen, R.A., Stone, P.J.: Classification and Regression Trees. Wadsworth, Belmont (1984)
5. Freund, Y., Mason, L.: The Alternating Decision Tree Learning Algorithm. In: Procedings of the 16th Int. Conference on Machine Learning, vol. 1, pp. 124–133. Morgan Kaufman, San Francisco (1999)
6. Freund, Y., Schapirem, R.E.: Experiments with a new boosting algorithm. In: Procedings of the 13th Int. Conference on Machine Learning, pp. 148–156. Morgan Kaufmann, San Francisco (1996)
7. Hájek, P., Havránek, T.: Mechanising Hypothesis Formation Mathematical Foundations for a General Theory. Springer, Heidelberg (1978)
8. Kohavi, R., Kunz, C.: Option Decision Trees with Majority Notes. In: Procedings of the 14th Int. Conference on Machine Learning, pp. 161–169. Morgan Kaufmann, San Francisco (1997)
9. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
10. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco (1993)
11. Ralbovský, M.: History and Future Development of the Ferda system. Mundus Symbolicus 15, 143–147 (2007)
12. Rauch, J., Šimůnek, M.: An Alternative Approach to Mining Association Rules. In: Proc. Foundations of Data Mining and Knowledge Discovery. Springer, Heidelberg (2005)
13. Šimůnek, M.: Academic KDD Project LISp-Miner. In: Advances in Soft Computing Intelligent Systems Design and Applications, vol. 272, pp. 263–272. Springer, Heidelberg (2003)
14. UCI Machine Learning Repository, http://www.ics.uci.edu/mlearn/MLRepository.html
15. Weka - Data Mining with Open Source Machine Learning Software, http://www.cs.waikato.ac.nz/ml/weka/

Mapping Data Mining Algorithms on a GPU Architecture: A Study

Ana Gainaru[1,2], Emil Slusanschi[1], and Stefan Trausan-Matu[1]

[1] University Politehnica of Bucharest, Romania
[2] University of Illinois at Urbana-Champaign, USA

Abstract. Data mining algorithms are designed to extract information from a huge amount of data in an automatic way. The datasets that can be analysed with these techniques are gathered from a variety of domains, from business related fields to HPC and supercomputers. The datasets continue to increase at an exponential rate, so research has been focusing on parallelizing different data mining techniques. Recently, GPU hybrid architectures are starting to be used for this task. However the data transfer rate between CPU and GPU is a bottleneck for the applications dealing with large data entries exhibiting numerous dependencies. In this paper we analyse how efficient data mining algorithms can be mapped on these architectures by extracting the common characteristics of these methods and by looking at the communication patterns between the main memory and the GPU's shared memory. We propose an experimental study for the performance of memory systems on GPU architectures when dealing with data mining algorithms and we also advance performance model guidelines based on the observations.

1 Introduction

1.1 Motivation

Data mining algorithms are generally used for the process of extracting interesting and unknown patterns or building models from any given dataset. Traditionally, these algorithms have their roots in the fields of statistics and machine learning. However, the amount of scientific data that needs to be analysed is approximately doubling every year [10] so the sheer volume of today's datasets is putting serious problems to the analysing process. Data mining is computationally expensive by nature and the size of the datasets that need to be analysed make the task even more expensive.

In recent years, there is an increasing interest in the research of parallel data mining algorithms. In parallel environments, algorithms are exploiting the vast aggregate main memory and processing power of parallel processors. During the last few years, Graphics Processing Units (GPU) have evolved into powerful processors that not only support typical computer graphics tasks but are also flexible enough to perform general purpose computations [9]. GPUs represent highly specialized architectures designed for graphics rendering, their development driven

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 102–112, 2011.

by the computer gaming industry. Recently these devices were successfully used to accelerate computationally intensive applications from a large variety of fields. The major advantage of today's GPUs is the combination they provide between extremely high parallelism and high bandwidth in memory transfer. GPUs offer floating point throughput and thousands of hardware thread contexts with hundreds of parallel compute pipelines executing programs in a SIMD fashion. High performance GPUs are now an integral part of every personal computer making this device very popular for algorithm optimizations.

However, it is not trivial to parallelize existing algorithms to achieve good performance as well as scalability to massive data sets on these hybrid architectures. First, it is crucial to design a good data organization and decomposition strategy so that the workload can be evenly partitioned among all threads with minimal data dependencies across them. Second, minimizing synchronization and communication overhead is crucial in order for the parallel algorithm to scale well. Workload balancing also needs to be carefully designed.

To best utilize the power computing resources offered by GPUs, it is necessary to examine to what extent traditionally CPU-based data mining problems can be mapped to a GPU architecture. In this paper, parallel algorithms specifically developed for GPUs with different types of data mining tasks are analysed. We are investigating how parallel techniques can be efficiently applied to data mining applications. Our goal in this paper is to understand the factors affecting GPU performance for these types of applications. We analyse the communication patterns for several basic data mining tasks and investigate what is the optimal way of dividing tasks and data for each type of algorithm.

1.2 Hardware Configuration

The GPU architecture is presented in Figure 1. The device has a different number of multiprocessors, where each is a set of 32-bit processors with a Single Instruction Multiple Data (SIMD) architecture. At each clock cycle, a multiprocessor executes the same instruction on a group of threads called a warp.

The GPU uses different types of memories. The shared memory (SM) is a memory unit with fast access and is shared among all processors of a multiprocessor. Usually SMs are limited in capacity and cannot be used for information which is shared among threads on different multiprocessors. Local and global memory reside in device memory (DM), which is the actual video RAM of the graphics card. The bandwidth for transferring data between DM and GPU is almost 10 times higher than that of CPU and main memory. So, a profitable way of performing computation on the device is to block data and computation to take advantage of fast shared memory by partition data into data subsets that fit into shared memory. The third kind of memory is the main memory which is not part of the graphics card. This memory is only accessed by the CPU so data needs to be transferred from one memory to another to be available for the GPU. The bandwidth of these bus systems is strictly limited, so these transfer operations are more expensive than direct accesses of the GPU to DM or of the CPU to main memory.

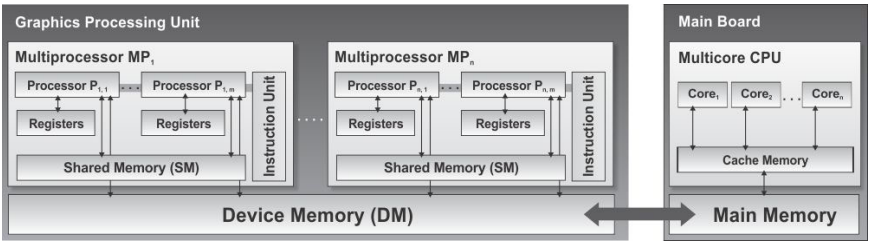


Fig. 1. GPU Architecture

The grapic processor used in our experimental study is a NVIDIA GeForce GT 420M. This processor works at 1GHz and consists of 12 Streaming Processors, each with 8 cores, making for a total of 96 cores. It features 2048MB device memory connected to the GPU via a 128-bit channel. Up to two data transfers are allowed to be made every clock cycle, so the peak memory bandwidth for this processor is 28.8GB/s. The computational power sums up to a peak performance of 134.4 GFLOP/s. The host machine is a Intel Core i3 370M processor at 2.4Ghz, with a 3MB L3 cache and 4GB of RAM. Nvidia offers a programming framework, CUDA, that allows the developer to write code for GPU with familiar C/C++ interfaces. Such frameworks model the GPU as a many-core architecture exposing hardware features for general-purpose computation. For all our tests we used NVIDIA CUDA 1.1.

The rest of the paper is organized as follows: Section 2 describes current parallelizations of different data mining algorithms that will be analysed, and presents results, highlighting the common properties and characteristics for them. Section 3 derives a minimal performance model for GPUs based on the factors discovered in the previous section. Finally, in section 4 we provide conclusions and present possible directions for future work.

2 Performance Study

2.1 Data Mining on GPUs

There are several algorithms proposed that implement different data mining algorithms for GPUs: classification [7,11], clustering [5,6], frequent itemset identification [1,2], association [3,4]. In this section we will give a short description for the ones that obtained the best speed-up result for every type. Since we are only interested in the most influential and widely used methods and algorithms in the data mining community, in this paper we investigate algorithms and methods that are described in the top 10 data mining algorithms paper [8].

Association rule mining is mostly represented by the Apriori and the Frequent Pattern Tree methods. Finding frequent item sets is not trivial because of its combinatorial explosion so there are many techniques proposed to parallelize both, the Apriori and the FP-growth algorithms, for parallel systems [12,13]. Based on those, more recently, research has started to focus on optimizing them

Table 1. Experimental datasets

Name	No. items	Avg. Length	Size	Density	Characteristics
dataset1	21.317	10	7M	7%	Synthetic
dataset2	87.211	10.3	25M	1.2%	Sparse/Synthetic
dataset3	73.167	53	41M	47%	Dense/Synthetic

for GPUs [1,2,3,4]. We analysed two of these methods, one for the Apriori and one for the FP-Growth methods.

The first method is [3] where the authors propose two implementations for the Apriori algorithm on GPUs. Both implementations exploit the bitmap representation of transactions, which facilitates fast set intersection to obtain transactions containing a particular item set. Both implementations follow the workflow of the original Apriori algorithm, one runs entirely on the GPU and eliminates intermediate data transfer between the GPU memory and the CPU memory while the other employs both the GPU and the CPU for processing. In [4] the authors propose a parallelization method for a cache-conscious FP-array. The FP-growth stage has a trivial parallelization, by assigning items to the worker threads in a dynamic manner. For building FP-trees, the authors divide the transactions into tiles, where different tiles contain the same set of frequent items. After this, the tiles are grouped and sent to the GPU. A considerable performance improvement is obtained due to the spatial locality optimization.

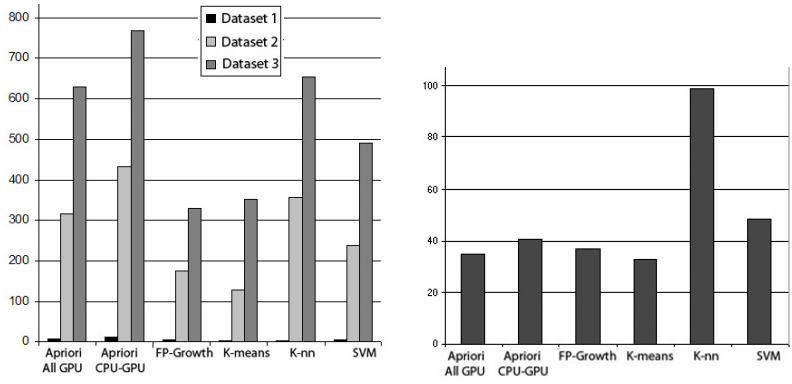
Most papers that deal with optimizing clustering methods are focusing on k-means methods [5,6] mainly because is the most parallel friendly algorithm. However, research is conducted on k-nn [14], neural networks [15], and density clustering [16]. Here we will analyze two of these methods. In [5] the clustering approach presented extends the basic idea of K-means by calculating simultaneously on GPU the distances from a single centroid to all objects at each iteration. In [14] the authors propose a GPU algorithm to compute the k-nearest neighbour problem with respect to Kullback-Leibler divergence [18]. Their algorithm's performance largely depends on the cache-hit ratio, and for a large data, it is likely that a cache miss occurs frequently.

Given an unsupervised learning technique, a classification algorithm builds a model that predicts whether a new example falls into one of the categories. [7] focuses on a SVM algorithm, used for building regression models in chemical informatics area. The SVM-light algorithm proposed implements various efficiency optimizations for GPUs to reduce the overall computational cost. The authors use a caching strategy to reuse previously calculated kernel values hence providing a good trade-off between memory consumption and training time.

Table 1 presents the datasets used for the experiments; one is small and fits in the GPU's shared memory and the others must be divided in subsets.

2.2 Memory Latency Analysis

We measure the latency of memory read and write operations and the execution time for different data mining algorithms. This experiment shows how much data



(a) Execution time (seconds)

(b) Execution time increase rate from dataset1 to dataset2

Fig. 2. Memory latency analysis

is exchanged by each algorithm from the main memory to the GPU's memory. Figure 2 presents the scaling obtained by each algorithm, for all considered datasets. As the first dataset fits entirely in the shared memory, it's execution time is accordingly fast, compared to the rest of the runs. Subsequently, the execution time increases with one order of magnitude for the first and second datasets, as is shown in figure 2(b). However, the second dataset is only three times larger. Since this is valid for the last two datasets as well, figure 2 is showing that, for all algorithms, the read latency for the input data once it does not fit in the shared memory is making the execution time increase dramatically. These numbers are showing that the way that each algorithm communicates with the input data influences the performance. The scalability limiting factors are largely from load imbalance and hardware resource contention.

In the second part, we investigate which part of each algorithm dominates the total execution time by running them on the scenarios proposed in their papers. We are especially interested to quantify how much of the total execution time is occupied by the data transfer between the main memory and the GPU memory when the input dataset does not fit in the shared memory.

Most of the algorithms have two main phases that alternate throughout the whole execution. For example, for both Apriori algorithms, apart for the time required by the data transfer between the GPU and CPU memory, candidate generation and support counting dominate the running time. Figure 3 shows the results for all the algorithms. The values presented represent the mean between the results from the second and third dataset. All methods need a significant percentage of time for data transfer, from 10% to over 30%. The datasets used for this experiment have the same size for all algorithms so that this difference is only given by the way in which the algorithms interact with the input data.

The Apriori method needs to pass through the entire dataset in each iteration. If the data does not fit in the GPU's memory it has to be transferred in pieces

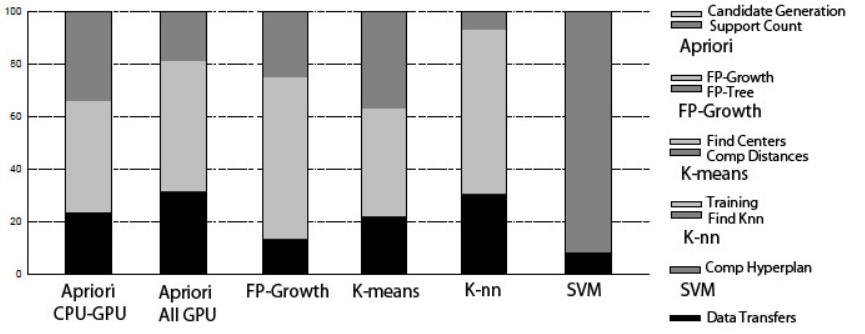


Fig. 3. Time breakdown (percentage)

every cycle. FP-Growth uses tiles with information that needs to be analysed together, so the communication time is better. K-means changes the centres in each iteration so the distances from all the points to the new centres must be computed in each iteration. Since the points are not grouped depending on how they are related, the communication time is also high. K-nn also needs to investigate the whole data set to find the closest neighbours, making it the algorithm with the minimal execution time per data size. SVM have mathematical tasks so the execution time for the same data is higher than in other cases. All these facts, together with the cache optimization strategy are the reason for the low communication latency.

The way in which algorithms interact with the input data set is also important – all need several passes through the entire datasets. Even though there are differences between how they interact with the data, there is one common optimization that could be applied to all: the input set could be grouped in a cache friendly way and transferred to the GPU's memory on a bulk of pieces. The rate at which the execution time increases is not necessary higher if the communication time is higher – this fact is best observed for the FP-growth method. This explains how well the algorithm scales. Thus, because the execution times for the Apriori and FP-Growth methods increase almost exponentially, even if the communication time for FP-Growth is not that high, the execution time increases considerably faster for a bigger dataset than for corresponding K-means implementation, since the equivalent K-means runs in less steps.

In the third experiment we investigate how much time each algorithm uses for CPU computations. Most of the algorithms are iterating through two different phases and need a synchronization barrier between them. Computations before and after each phase are made on the CPU. Figure 4 presents the percentage of the total execution that is used for CPU processing. Each algorithm is exploiting thread-level parallelism to fully take advantage of the available multi-threading capabilities, thus minimizing the CPU execution time. However the mean for all algorithms is still between 15% and 20% so it is important to continue to optimize this part as well. In this experiment we did not take into consideration the pre and post-processing of the initial dataset and generated results.

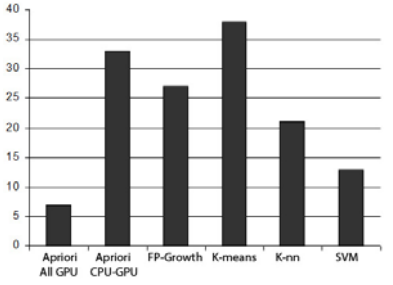
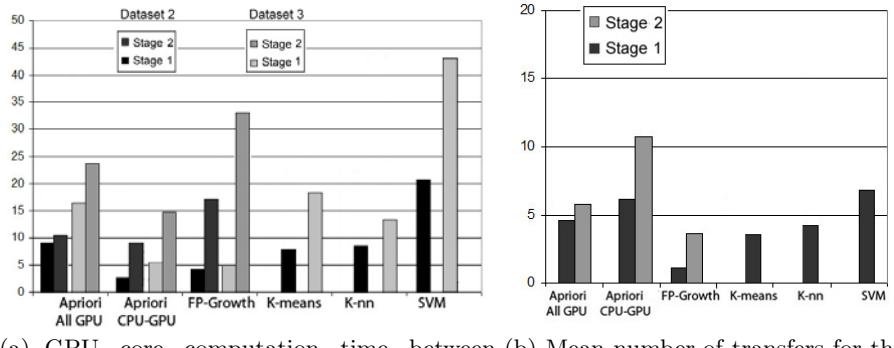


Fig. 4. CPU computation time (percentage)

2.3 Communication Patterns

We furthermore investigate the source code for all algorithms and extracted the communication patterns between the main memory and the GPU's shared memory. We are only interested in analysing the datasets that do not fit in the shared memory. All algorithms consist of a number of iterations over same phases. We investigate what is the amount of work done by each GPU core between two memory transfers and how many times the same data is transferred to the GPU's memory in one iteration. The results are presented in Figure 5.

Apriori's first stage, the candidate generation, alternates data transfers towards the GPU's memory with computational phases. At the end of each phase, the list of candidates is sent to the CPU, and stage two begins – the count support. For both phases, the same data is investigated multiple times and the amount of work for it is low. The FP-Growth's first phase alternates both transfers to and from the shared memory between computations. The amount of work for each data is not very high either, however here tiles are formed to optimize the memory access, so we observe the best results being delivered by this algorithm.



(a) GPU core computation time between transfers for each phase

(b) Mean number of transfers for the same data for each phase

Fig. 5. Communication patterns

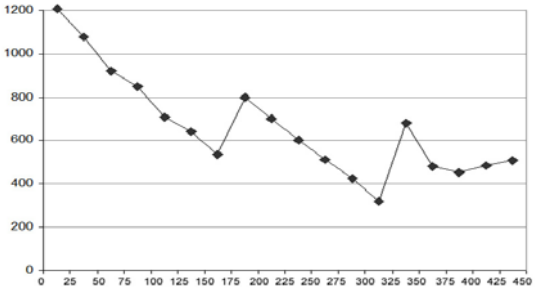


Fig. 6. Execution time variation with the increase of threads in a block

The K-means and K-nn algorithms have one part done on the GPU, finding distances from all the points to the centres and finding the k nearest neighbours. The algorithms have no strategy that is cache friendly. Even if the same data is transferred a few times in one iteration, the amount of work is still very low. SVM has mathematical computations for each data making it efficient for optimization on GPU's. However the same data is transferred multiple times. The authors implemented a memory optimization protocol making it a very efficient solution. We observed that, if each time a thread in a core needs some data, the algorithm transfers it to the GPU, leading to a higher total communication latency. From a performance point of view, it seems that it is better to gather data in bulks that need to be analysed together and send them all at once, even if this means that some threads will be blocked waiting for the data.

2.4 Performance Analysis for Different Number of Threads

We investigate how the average read latency is affected by the number of threads that are being executed. To this end we launch the applications with a constant grid size and change the number of threads in a block. The GPU executes threads as groups in a warp, so the average latency experienced by a single thread does not increase with the increase in the number of threads.

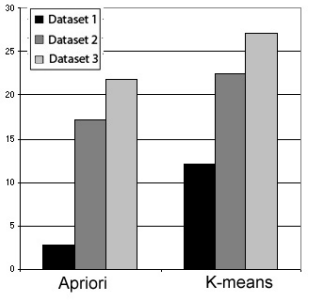
We only analyse the datasets that exceed the shared memory. All algorithms behave in the same way as can be seen in Figure 6. The algorithm's performance has shifts at some points because it is dependent on the number of threads available in each Steaming Multiprocessor and not on the total number of threads, as is explained in [17]. The performance keeps increasing until we obtain different points for each algorithm. Finally there is not enough space left for all threads to be active. The exact values depend on the data mining method under investigation and on the GPU parameters, so this information cannot be included in the model.

3 Performance Optimization Guidelines

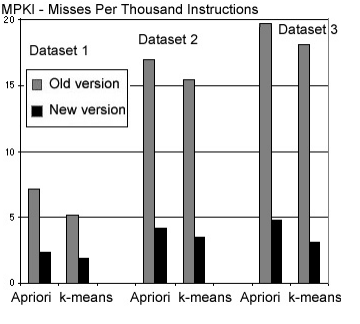
We present a model that offers optimization for any data mining algorithms at the CPU and GPU level. The first one minimizes the L3 cache misses by

pre-loading the next several data points in the active set beforehand according to their relation with the current investigated data. The second one maximizes the amount of work done by the GPU cores for each data transfer. We analyse two data mining techniques, namely the Apriori and K-means, and we group the data sent to the GPU at each iterations in bulks that we send together even if we risk keeping some threads waiting for information. We show that by designing carefully the memory transfers, both from memory to caches or to and from the GPU's memory, all data mining algorithms can be mapped very well on hybrid computing architectures. All these methods have relatively low computations per unit of data so the main bottleneck in this case is memory management. The model is thus composed by three tasks: to increase the core execution time per data transfer – the increase must be for each core and not for each thread; to eliminate synchronization by trying to merge all steps in each iteration; and to optimize the CPU computation time.

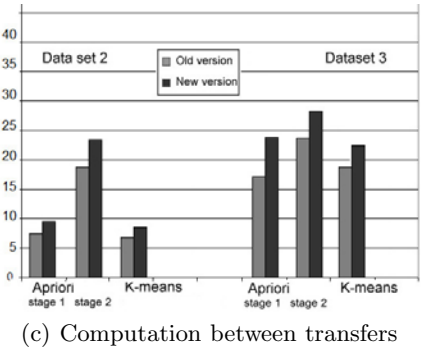
Our method preloads the next several transactions in the item set for the Apriori method, or next points for K-means beforehand according to when they need to be analysed. This influences both the L3 cache miss rate and how many times the same data is sent from one memory to another. The performance impact of memory grouping and prefetching is thus more evident for large datasets, with many dimensions for K-means and longer transaction length for Apriori,



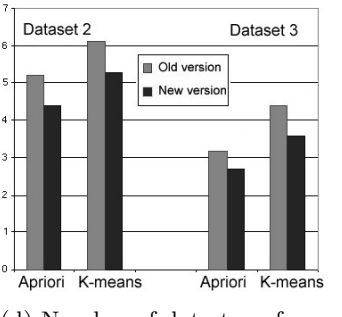
(a) Performance speedup



(b) L3 cache miss reduction



(c) Computation between transfers



(d) Number of data transfers

Fig. 7. Optimization results

because the longer the dimension of the transaction the fewer points are installed in the cache, so there are more opportunities for prefetching. We therefore synchronize data only between two iterations by merging the computation for the two phases in each cycle.

Figure 7 plots the performance increase, the L3 cache misses, the amount of work and data transaction for each iteration. The implementation reduces the data transaction by a factor of 13.6 on average and provides 17% increase in computation time per block of data transferred to the GPU. The group creation improves the temporal data locality performance and reduces the cache misses by a factor of 32.1 on average. Even if dataset grouping increases the pre-processing stage, the overall performance of both algorithms has improved with 20% compared to the methods presented in the previous sections.

4 Conclusion and Future Work

In this paper we analyse different data mining methods and present a view for how much improvement might be possible with GPU acceleration on these techniques. We extract the common characteristics for different clustering, classification and association extraction methods by looking at the communication pattern between the main memory and GPU's shared memory. We present experimental studies for the performance of the memory systems on GPU architectures by looking at the read latency and the way the methods interact with the input dataset. We presented performance optimization guidelines based on the observations and manually implemented the modification on two of the algorithms. The observed performance is encouraging, so for the future we plan to develop a framework that can be used to automatically parallelize data mining algorithms based on the proposed performance model.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: International Conference on Very Large Data Bases, pp. 487–499 (1994)
2. Han, J., et al.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery 8(1) (2004)
3. Fang, W., et al.: Wenbin Fang and all: Frequent Itemset Mining on Graphics Processors (2009)
4. Liu, L., et al.: Optimization of Frequent Itemset Mining on Multiple-Core Processor. In: International Conference on Very Large Data Bases, pp. 1275–1285 (2007)
5. Shalom, A., et al.: Efficient k-means clustering using accelerated graphics processors. In: International Conference on Data Warehousing and Knowledge Discovery, pp. 166–175 (2008)
6. Cao, F., Tung, A.K.H., Zhou, A.: Scalable clustering using graphics processors. In: Yu, J.X., Kitsuregawa, M., Leong, H.-V. (eds.) WAIM 2006. LNCS, vol. 4016, pp. 372–384. Springer, Heidelberg (2006)
7. Liao, Q., et al.: Accelerated Support Vector Machines for Mining High-Throughput Screening Data. J. Chem. Inf. Model. 49(12), 2718–2725 (2009)

8. Wu, X., et al.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1) (2007)
9. Lastra, A., Lin, M., Manocha, D.: Gpgp: General purpose computation using graphics processors. In: ACM Workshop on General Purpose Computing on Graphics Processors (2004)
10. Li, J., et al.: Parallel Data Mining Algorithms for Association Rules and Clustering. In: International Conference on Management of Data (2008)
11. Carpenter, A.: CuSVM A cuda implementation of support vector classification and regression (2009), http://patternsonascreen.net/cuSVM.html
12. Pramudiono, I., et al.: Tree structure based parallel frequent pattern mining on PC cluster. In: International Conference on Database and Expert Systems Applications, pp. 537–547 (2003)
13. Pramudiono, I., Kitsuregawa, M.: Tree structure based parallel frequent pattern mining on PC cluster. In: Mařík, V., Štěpánková, O., Retschitzegger, W. (eds.) DEXA 2003. LNCS, vol. 2736, pp. 537–547. Springer, Heidelberg (2003)
14. Garcia, V., et al.: Fast k nearest neighbor search using GPU. In: Computer Vision and Pattern Recognition Workshops (2008)
15. Oh, K.-S., et al.: GPU implementation of neural networks. Journal of Pattern Recognition 37(6) (2004)
16. Domeniconi, C., et al.: An Efficient Density-based Approach for Data Mining Tasks. Journal of Knowledge and Information Systems 6(6) (2004)
17. Domeniconi, C., et al.: OpenMP to GPGPU: a compiler framework for automatic translation and optimization. In: Symposium on Principles and Practice of Parallel Programming, pp. 101–110 (2009)
18. Wang, Q.: Divergence estimation of continuous distributions based on data-dependent partitions. IEEE Transactions on Information Theory, 3064–3074 (2005)

Applying Domain Knowledge in Association Rules Mining Process – First Experience*

Jan Rauch and Milan Šimůnek

Faculty of Informatics and Statistics, University of Economics, Prague
nám W. Churchilla 4, 130 67 Prague 3, Czech Republic
{rauch,simunek}@vse.cz

Abstract. First experiences with utilization of formalized items of domain knowledge in a process of association rules mining are described. We use association rules - atomic consequences of items of domain knowledge and suitable deduction rules to filter out uninteresting association rules. The approach is experimentally implemented in the LISp–Miner system.

1 Introduction

One of great challenges in data mining research is application of domain knowledge in data mining process [3]. Our goal is to present first experiences with an approach to use domain knowledge in association rules mining outlined in [5]. We deal with association rules of the form $\varphi \approx \psi$ where φ and ψ are Boolean attributes derived from columns of an analyzed data matrix. An example of data matrix is in section 2. Not only conjunctions of *attribute-value* pairs but general Boolean expressions built from *attribute-set of values* pairs can be used. Symbol $\approx$ means a general relation of φ and ψ, see section 3.

We deal with formalized items of domain knowledge related to analyzed domain knowledge, see section 2. We apply the 4ft-Miner procedure for mining association rules. It deals with Boolean expressions built from *attribute-set of value*. An example of an analytical question solution of which benefits from properties of 4ft-Miner is in section 4.

The paper focuses on problem of filtering out of association rules which can be considered as consequences of given items of domain knowledge as suggested in [5]. Our approach is based on mapping of each item of domain knowledge to a suitable set of association rules and also on deduction rules concerning pairs of association rules. The approach is implemented in the LISp-Miner system which involves also the 4ft-Miner procedure. An example of its application is also in section 4. It can result in finding of interesting exceptions from items of domain knowledge in question, but the way of dealing with exceptions differs from that described in [8].

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

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 113–122, 2011.

2 STULONG Data Set

2.1 Data Matrix Entry

We use data set STULONG concerning *Longitudinal Study of Atherosclerosis Risk Factors*[1]. Data set consists of four data matrices, we deal with data matrix *Entry* only. It concerns 1 417 patients – men that have been examined at the beginning of the study. Each row of data matrix describes one patient. Data matrix has 64 columns corresponding to particular attributes – characteristics of patients. The attributes can be divided into various groups, We use three groups defined for this paper - *Measurement*, *Difficulties*, and *Blood pressure*.

Group *Measurement* has three attributes - *BMI* i.e. Body Mass Index, *Subsc* i.e. skinfold above musculus subscapularis (in mm), and *Tric* i.e. skinfold above musculus triceps (in mm). The original values were transformed such that these attributes have the following possible values (i.e. categories):
BMI : $(16;21\rangle,\ (21;22\rangle,\ (22;23\rangle,\ \ldots,\ (31;32\rangle, > 32$ (13 categories)
Subsc : $\langle 4;10),\ \langle 10;12),\ \langle 12;14),\ \ldots,\ \langle 30;32),\ \langle 32;36), > 36$ (14 categories)
Tric : $1-4,\ 5, 6, \ldots, 12, 13-14, 15-17, \geq 18$ (12 categories).

Group *Difficulties* has three attributes with 2 - 5 categories, frequencies of particular categories are in brackets (there are some missing values, too):
Asthma with 2 categories: *yes* (frequency 1210) and *no* (frequency 192)
Chest i.e. *Chest pain* with 5 categories: *not present* (1019), *non ischaemic* (311), *angina pectoris* (52), *other* (19), *possible myocardial infarction* (3)
Lower limbs i.e. *Lower limbs pain* with 3 categories: *not present* (1282), *non ischaemic* (113), *claudication* (17).

Group *Blood pressure* has two attributes - *Diast* i.e. Diastolic blood pressure and *Syst* i.e. Systolic blood pressure The original values were transformed such that these attributes have the following categories:
Diast : $\langle 45;65),\ \langle 65;75),\ \langle 75;85),\ \ldots,\ \langle 105;115), > 115$ (7 categories)
Syst : $\langle 85;105),\ \langle 105;115),\ \langle 115;125),\ \ldots,\ \langle 165;175), > 175$ (9 categories).

2.2 Domain Knowledge

There are various types of domain knowledge related to STULONG data. Three of them in a formalized form are managed by the LISp-Miner system [7]: *groups of attributes*, *information on particular attributes* and *mutual influence of attributes*.

There are 11 basic groups (see `http://euromise.vse.cz/challenge2004/data/entry/`. These groups are mutually disjoint and their union is the set of

[1] The study (STULONG) was realized at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and University Hospital in Prague, under the supervision of Prof. F. Boudík, MD, DSc., with collaboration of M. Tomečková, MD, PhD and Prof. J. Bultas, MD, PhD. The data were transferred to the electronic form by the European Centre of Medical Informatics, Statistics and Epidemiology of Charles University and Academy of Sciences CR(head. Prof. J. Zvárová, PhD, DSc.). The data resource is on the web pages `http://euromise.vse.cz/challenge2004/`.

all attributes. We call these groups *basic groups of attributes*, they are perceived by physicians as reasonable sets of attributes. It is also possible to define additional groups of atributes for some specific tasks, see e.g. groups *Measurement*, *Difficulties*, and *Blood pressure* introduced above.

Examples of information on particular attributes are boundaries for classification of overweight and obesity by BMI. Overweight is defined as BMI in $25.0 - 29.9$ and obesity as $BMI \geq 30$.

There are several types of influences among attributes. An example is expression $BMI \uparrow\uparrow Diast$ saying that if body mass index of patient increases then its diastolic blood pressure increases too.

3 Association Rules

The association rule is understood to be an expression $\varphi \approx \psi$ where φ and ψ are Boolean attributes. It means that the Boolean attributes φ and ψ are associated in the way given by the symbol $\approx$. This symbol is called the *4ft-quantifier*. It corresponds to a condition concerning a four-fold contingency table of φ and ψ. Various types of dependencies of φ and ψ can be expressed by 4ft-quantifiers.

The association rule $\varphi \approx \psi$ concerns analyzed data matrix $\mathcal{M}$. An example of a data matrix is data matrix *Entry* a fragment of which is in figure 1.

	attributes			examples of basic Boolean attributes	
patient	*Asthma*	*BMI*	...	*Asthma*(*yes*)	*BMI* $((21;22\rangle,(22;23\rangle)$
o_1	*yes*	$(16;21\rangle$	...	1	0
⋮	⋮	⋮	⋱	⋮	⋮
o_{1417}	*no*	$(22;23\rangle$	...	0	1

Fig. 1. Data matrix $\mathcal{M}$ and examples of Boolean attributes

The Boolean attributes are derived from the columns of data matrix $\mathcal{M}$. We assume there is a finite number of possible values for each column of $\mathcal{M}$. Possible values are called *categories*. *Basic Boolean attributes* are created first. The basic Boolean attribute is an expression of the form $A(\alpha)$ where $\alpha \subset \{a_1, \ldots a_k\}$ and $\{a_1, \ldots a_k\}$ is the set of all possible values of the column A. The basic Boolean attribute $A(\alpha)$ is true in row o of $\mathcal{M}$ if it is $a \in \alpha$ where a is the value of the attribute A in row o. Set α is called a *coefficient* of $A(\alpha)$. Boolean attributes are derived from basic Boolean attributes using propositional connectives $\vee$, $\wedge$ and $\neg$ in a usual way.

There are two examples of basic Boolean attributes in figure 1 - $Asthma(yes)$ and $BMI((21;22\rangle,(22;23\rangle)$. Attribute $Asthma(yes)$ is true for patient o_1 and false for patient o_{1417}, we write "1" or "0" respectively. Attribute $BMI((21;22\rangle,(22;23\rangle)$ is false for o_1 because of $(16;21\rangle \notin \{(21;22\rangle,(22;23\rangle)\}$ and true for o_{1417} because of $(22;23\rangle \in \{(21;22\rangle,(22;23\rangle)\}$. Please note that we should write $Asthma(\{yes\})$ etc. but we will not do it. We will also usually write $BMI(21;23\rangle$ instead of $BMI\ ((21;22\rangle,(22;23\rangle)$ etc.

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

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

The rule $\varphi \approx \psi$ is *true in data matrix* $\mathcal{M}$ if the condition corresponding to the 4ft-quantifier is satisfied in the four-fold contingency table of φ and ψ in $\mathcal{M}$, otherwise $\varphi \approx \psi$ is *false in data matrix* $\mathcal{M}$. The four-fold contingency table $4ft(\varphi, \psi, \mathcal{M})$ of φ and ψ in data matrix $\mathcal{M}$ is a quadruple $\langle a, b, c, d\rangle$ where a is the number of rows of $\mathcal{M}$ satisfying both φ and ψ, b is the number of rows of $\mathcal{M}$ satisfying φ and not satisfying ψ etc., see Table 1.

There are various 4ft-quantifiers, some of them are based on statistical hypothesis tests, see e.g. [1,6]. We use here a simple 4ft-quantifier i.e. quantifier $\Rightarrow_{p,Base}$ of *founded implication* [1]. It is defined for $0 < p \leq 1$ and $Base > 0$ by the condition $\frac{a}{a+b} \geq p \ \wedge \ a \geq Base$. The association rule $\varphi \Rightarrow_{p,Base} \psi$ means that at least $100p$ per cent of rows of $\mathcal{M}$ satisfying φ satisfy also ψ and that there are at least $Base$ rows of $\mathcal{M}$ satisfying both φ and ψ. We use this quantifier not only because of its simplicity but also because there are known deduction rules related to this quantifier [4].

4 Applying LISp-Miner System

The goal of this paper is to describe an application of an approach to filtering out association rules, which can be considered as consequences of given items of domain knowledge. This approach is based on mapping of items of domain knowledge in question to suitable sets of association rules and also on deduction rules concerning pairs of association rules. We deal with items of domain knowledge stored in the LISp-Miner system outlined in section 2.2. We use GUHA procedure 4ft-Miner [6] which mines for association rules described in section 3. In addition we outline how the groups of attributes can be used to formulate reasonable analytical questions.

An example of a reasonable analytical question is given in section 4.1. Input of the 4ft-Miner procedure consists of parameters defining a set of relevant association rules and of an analyzed data matrix. Output consists of all relevant association rules true in input data matrix. There are fine tools to define set of association rules which are relevant to the given analytical question. We use data *Entry*, see section 2.1. Input parameters of 4ft-Miner procedure suitable to solve our analytical question are described also in section 4.1. There are 158 true relevant association rules found for these parameters.

Our analytical question is formulated such that we are not interested in consequences of item of domain knowledge *BMI* $\uparrow\uparrow$ *Diast*. This item says that if body mass index of patient increases then his diastolic blood pressure increases too, see section 2.2. However, there are many rules among 158 resulting rules which can be considered as consequences of item *BMI* $\uparrow\uparrow$ *Diast*. We filter out these consequences in two steps.

In the first step we define a set $Cons(BMI \uparrow\uparrow Diast, Entry, \approx)$ of atomic consequences of $BMI \uparrow\uparrow Diast$. Each atomic consequence is an association rule of the form $BMI(\omega) \approx Diast(\delta)$ which can be considered as true in data matrix *Entry* if $BMI \uparrow\uparrow Diast$ is supposed to be true. In addition, $\approx$ is a 4ft-quantifier used in the 4ft-Miner application in question. For more details see section 4.2.

In the second step we filter out each association rule $\varphi \approx \psi$ from the output of 4ft-Miner which is equal to an atomic consequence or can be considered as a consequence of an atomic consequence. There are additional details in section 4.3.

4.1 Analytical Question and 4ft-Miner

Let us assume we are interested in an analytical question:

Are there any interesting relations between attributes from group Measurement *and attributes from group* Blood pressure *in the data matrix* Entry*? Attributes from group* Measurement *can be eventually combined with attributes from group* Difficulties*. Interesting relation is a relation which is strong enough and which is not a consequence of the fact* BMI $\uparrow\uparrow$ Diast.

This question can be symbolically written as

$$Measurement \wedge Difficulties \longrightarrow Blood\ pressure \quad [Entry\,;\, BMI \uparrow\uparrow Diast]\ .$$

We deal with association rules, thus we convert our question to a question concerning association rules. Symbolically we can express a converted question as

$$\mathcal{B}[Measurement] \wedge \mathcal{B}[Difficulties] \ \approx\ \mathcal{B}[Blood\ pressure] \quad [Entry\,;\, BMI \uparrow\uparrow Diast]\ .$$

Here B[*Measurement*] means a set of all Boolean attributes derived from attributes of the group *Measurement* we consider relevant to our analytical question, similarly for $\mathcal{B}[Difficulties]$ and $\mathcal{B}[Blood\ pressure]$.

We search for rules $\varphi_M \wedge \varphi_D \ \approx\ \psi_B$ which are true in data matrix *Entry*, cannot be understood as a consequence of $BMI \uparrow\uparrow Diast$ and $\varphi_M \in \mathcal{B}[Measurement]$, $\varphi_D \in \mathcal{B}[Difficulties]$, and $\psi_B \in \mathcal{B}[Blood\ pressure]$.

The procedure 4ft-Miner does not use the well known a-priori algorithm. It is based on representation of analyzed data by suitable strings of bits [6]. That's way 4ft-Miner has very fine tools to define such set of association rules. One of many possibilities how to do it is in figure 2. Remember that we deal with rules $\varphi \approx \psi$, φ is called *antecedent* and ψ is *succedent*. Set B[*Measurement*] is defined in column `ANTECEDENT` in row `Measurement Conj, 1-3` and in three consecutive rows.

Each φ_M is a conjunction of 1 - 3 Boolean attributes derived from particular attributes of the group *Measurement*. Set of all such Boolean attributes derived from attribute *BMI* is defined by the row `BMI(int), 1-3 B, pos`. It means that all Boolean attributes $BMI(\alpha)$ where α is a set of 1 - 3 consecutive categories (i.e. interval of categories) are generated. Examples of such Boolean attributes are $BMI(16;21\rangle)$, $BMI((21;22\rangle, (22;23\rangle)$ i.e. $BMI(21;23\rangle$, and $BMI((21;22\rangle, (22;23\rangle, (23;24\rangle)$ i.e. $BMI(21;24\rangle$. Sets of Boolean attributes derived from attributes *Subsc* and *Tric* are defined similarly. An example of $\varphi_M \in \mathcal{B}[Measurement]$ is conjunction $\varphi_M = BMI(21;24\rangle \wedge Subsc\langle 4;14)$.

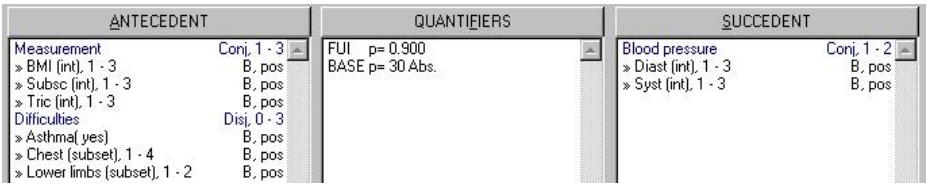


Fig. 2. Input parameters of the 4ft-Miner procedure

Each φ_D is a disjunction of 0 - 3 Boolean attributes derived from particular attributes of the group *Difficulties*. There is only one Boolean attribute derived from attribute *Asthma* i.e. *Asthma*(*yes*). Set of all such Boolean attributes derived from attribute *Chest* is defined by the row `Chest(subset), 1-4 B, pos`. It means that all Boolean attributes *Chest*(α) where α is a subset of 1 - 4 categories of attribute *Chest* are generated. In addition, category *not present* is not taken into account (not seen in figure 2). Similarly, all Boolean attributes *Lower limbs*(α) where α is a subset of 1 - 2 categories are generated and category *not present* is not considered. Please note, that a disjunction of zero Boolean attributes means that φ_D is not considred.

Set $\mathcal{B}$[*Blood pressure*] is defined in row `Blood pressure Conj, 1-2` of column `SUCCEDENT` and in two consecutive rows in a way similar to that in which set $\mathcal{B}$[*Measurement*] is defined. In column `QUANTIFIERS` the quantifier $\Rightarrow_{0.9,30}$ of founded implication is specified.

This task was solved in 171 minutes (PC with 2GB RAM and Intel T7200 processor at 2 GHz). $456 * 10^6$ association rules were generated and tested, 158 true rules were found. The rule $BMI(21;22\rangle \wedge Subsc(< 14) \Rightarrow_{0.97,33} Diast\langle 65;75)$ is the strongest one (i.e. with highest confidence). It means that 34 patients satisfy $BMI(21;22\rangle \wedge Subsc(< 14)$ and 33 from them satisfy also $Diast\langle 65;75)$.

Most of found rules have attribute *BMI* in antecedent and attribute *Diast* in succedent (as the above shown rule). We can expect that lot of such rules can be seen as a consequences of *BMI* $\uparrow\uparrow$ *Diast*.

4.2 Atomic Consequences of *BMI* $\uparrow\uparrow$ *Diast*

We define a set $Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$ of simple rules in the form $BMI(\omega) \approx Diast(\delta)$ which can be considered as consequences of *BMI* $\uparrow\uparrow$ *Diast*. Such rules are called *atomic consequences of* BMI $\uparrow\uparrow$ Diast. We assume that this set is usually defined by a domain expert.

Examples of such atomic consequences are rules $BMI(low) \Rightarrow_{0.9,30} Diast(low)$ saying that at least 90 per cent of patients satisfying $BMI(low)$ satisfy also $Diast(low)$ and that there are at least 30 patients satisfying both $BMI(low)$ and $Diast(low)$. The only problem is to define suitable coefficients *low* for both attributes *BMI* and *Diast*.

Let us remember that there are 13 categories of *BMI* - $(16;21\rangle$, $(21;22\rangle$, $(21;22\rangle$, ..., $(31;32\rangle$, > 32 and 7 categories of *Diast* - $\langle 45;65)$, $\langle 65;75)$, ...,

$\langle 105; 115)$, > 115. We can decide that each Boolean attribute $BMI(\omega)$ where $\omega \subset \{(16; 21\rangle, (21; 22\rangle, (21; 22\rangle\}$ will be considered as $BMI(low)$ (we use low quarter of all categories) and similarly each Boolean attribute $Diast(\delta)$ where $\delta \subset \{\langle 45; 65), \langle 65; 75)\}$ will be considered as $Diast(low)$ (we use low third of all categories). We can say that rules $BMI(low) \Rightarrow_{0.9,30} Diast(low)$ are defined by a rectangle $\mathcal{A}_{low} \times \mathcal{S}_{low}$ = Antecedent $\times$ Succedent where

$$\text{Antecedent} \times \text{Succedent} = \{(16; 21\rangle, (21; 22\rangle, (21; 22\rangle\} \times \{\langle 45; 65), \langle 65; 75)\}$$

There is *LMDataSource* module in the LISp-Miner system which makes possible to do various input data transformations and in addition it also allows to define the set $Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$ as a union of several similar, possibly overlapping, rectangles $\mathcal{A}_1 \times \mathcal{S}_1, \ldots, \mathcal{A}_R \times \mathcal{S}_R$ such that $BMI(\omega) \Rightarrow_{0.9,30} Diast(\delta) \in Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$ if and only if there is an $i \in \{1, \ldots, K\}$ such that $\omega \subseteq \mathcal{A}_i$ and $\delta \subseteq \mathcal{S}_i$. An example of definition of a set $Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$ is in figure 3, six rectangles are used.

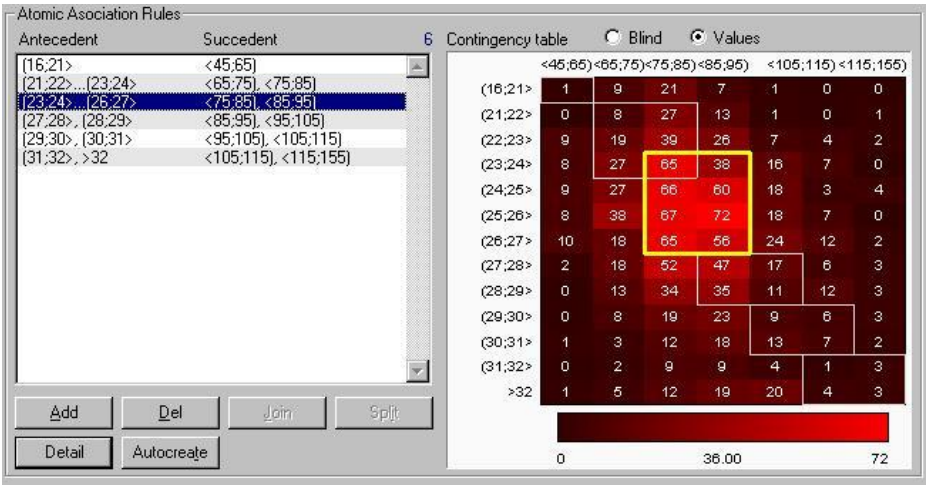


Fig. 3. Definition of atomic association rules

4.3 Filtering Out Consequences of *BMI* ↑↑ *Diast*

We will discuss possibilities of filtering out all rules from the output rules which can be considered as consequences of given item of domain knowledge. We take into account both strict logical deduction – see (ii), and specific conditions also supporting filtering out additional rules – see (iii). We use the set $Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$ of atomic consequences $BMI(\omega) \Rightarrow_{0.9,30} Diast(\delta)$ of $BMI \uparrow\uparrow Diast$ defined in figure 3. We filter out each of 158 output rules $\varphi \Rightarrow_{0.9,30} \psi$ satisfying one of conditions (i), (ii), (iii):

$\mathcal{M}$	$Diast\langle 65;85)$	$\neg Diast\langle 65;85)$
$BMI(21;22\rangle$	a	b
$\neg BMI(21;22\rangle$	c	d

$\mathcal{M}$	$Diast\langle 65;95)$	$\neg Diast\langle 65;95)$
$BMI(21;22\rangle$	a'	b'
$\neg BMI(21;22\rangle$	c'	d'

Fig. 4. $4ft(BMI(21;22\rangle, Diast\langle 65;85), \mathcal{M})$ and $4ft(BMI(21;22\rangle, Diast\langle 65;95), \mathcal{M})$

(i) $\varphi \Rightarrow_{0.9,30} \psi$ is equal to an atomic consequence $BMI(\omega) \Rightarrow_{0.9,30} Diast(\delta)$

(ii) $\varphi \Rightarrow_{0.9,30} \psi$ is a logical consequence of an atomic consequence $BMI(\omega) \Rightarrow_{0.9,30} Diast(\delta)$

(iii) $\varphi \Rightarrow_{0.9,30} \psi$ is in the form $\varphi_0 \wedge \varphi_1 \Rightarrow_{0.9,30} \psi_0$ where $\varphi_0 \Rightarrow_{0.9,30} \psi_0$ satisfies (i) or (ii). We filter out such rules because of patients satisfying $\varphi_0 \wedge \varphi_1$ satisfy also φ_0 and thus the rule $\varphi_0 \wedge \varphi_1 \Rightarrow_{0.9,30} \psi_0$ does not say something new in comparison with $\varphi_0 \Rightarrow_{0.9,30} \psi_0$ even if its confidence is higher than 0.9.

We give more details below.

(i): There is no atomic consequence $BMI(\omega) \Rightarrow_{0.9,30} Diast(\delta)$ belonging to $Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$ among the output rules.

(ii): There is output rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;95)$ not belonging to $Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$. Rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;85)$ belongs to $Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$, see second row in the left part of figure 3. In addition, rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;95)$ logically follows from rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;85)$.

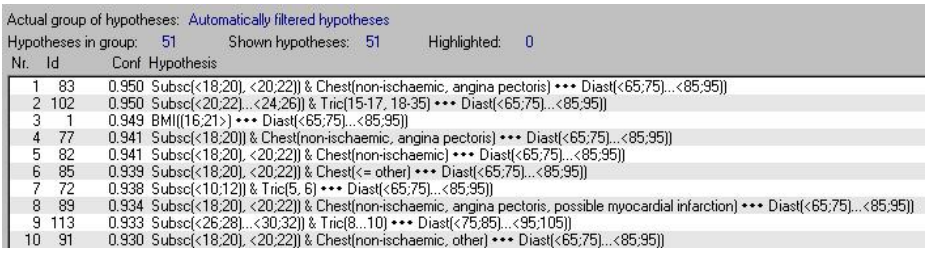
It means that if rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;85)$ is true in a given data matrix $\mathcal{M}$ then rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;95)$ is true in $\mathcal{M}$ too. Rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;85)$ is for data matrix *Entry* considered as a consequence of $BMI \uparrow\uparrow Diast$ and thus rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;95)$ can also be considered as a consequence of $BMI \uparrow\uparrow Diast$ for data matrix *Entry*. It means that rule $BMI(21;22\rangle \Rightarrow_{0.97,30} Diast\langle 65;95)$ is filtered out.

We demonstrate why rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;95)$ logically follows from rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;85)$. In figure 4 there are 4ft-tables $4ft(BMI(21;22\rangle, Diast\langle 65;85), \mathcal{M})$ of $BMI(21;22\rangle$ and $Diast\langle 65;85)$ in $\mathcal{M}$ and $4ft(BMI(21;22\rangle, Diast\langle 65;95), \mathcal{M})$ of $BMI(21;22\rangle$ and $Diast\langle 65;95)$ in $\mathcal{M}$. It is clear that $a + b = a' + b'$. In addition each patient satisfying $Diast\langle 65;85)$ satisfy also $Diast\langle 65;95)$ and thus $a' \geq a$ and $b' \leq b$ which means that if $\frac{a}{a+b} \geq 0.9 \wedge a \geq 30$ then also $\frac{a'}{a'+b'} \geq 0.9 \wedge a' \geq 30$.

Note that there is a theorem proved in [4] which makes possible to easy decide if association rule $\varphi' \Rightarrow_{p,Base} \psi'$ logically follows from $\varphi \Rightarrow_{p,Base} \psi$ or not.

(iii) It is also easy to show that rule $BMI(21;22\rangle \wedge Subsc(\leq 14) \Rightarrow_{0.9,30} Diast\langle 65;95)$ does not logically follow from rule $BMI(21;22\rangle \Rightarrow_{0.9,30} Diast\langle 65;95)$. However, patients satisfying $BMI(21;22\rangle \wedge Subsc(\leq 14)$ satisfy also $BMI(21;22\rangle$ and thus rule $BMI(21;22\rangle \wedge Subsc(\leq 14) \Rightarrow_{0.9,30} Diast\langle 65;95)$ does not say something new and can be also filtered out. (This could be of course a subject of additional discussion, however we will not discuss here due to limited space.)

From the same reason we filter out each rule $BMI(\rho) \wedge \varphi_1 \Rightarrow_{0.9,30} Diast(\tau)$ if the rule $BMI(\rho) \Rightarrow_{0.9,30} Diast(\tau)$ satisfies (i) or (ii). After filtering out all rules



Actual group of hypotheses: Automatically filtered hypotheses

Hypotheses in group: 51 Shown hypotheses: 51 Highlighted: 0

Nr.	Id	Conf	Hypothesis
1	83	0.950	Subsc(<18;20), <20;22)) & Chest(non-ischaemic, angina pectoris) ••• Diast(<65;75)...<85;95))
2	102	0.950	Subsc(<20;22)...<24;26)) & Tric(15-17, 18-35) ••• Diast(<65;75)...<85;95))
3	1	0.949	BMI([16;21>) ••• Diast(<65;75)...<85;95))
4	77	0.941	Subsc(<18;20)) & Chest(non-ischaemic, angina pectoris) ••• Diast(<65;75)...<85;95))
5	82	0.941	Subsc(<18;20), <20;22)) & Chest(non-ischaemic) ••• Diast(<65;75)...<85;95))
6	85	0.939	Subsc(<18;20), <20;22)) & Chest(<= other) ••• Diast(<65;75)...<85;95))
7	72	0.938	Subsc(<10;12)) & Tric(5, 6) ••• Diast(<65;75)...<85;95))
8	89	0.934	Subsc(<18;20), <20;22)) & Chest(non-ischaemic, angina pectoris, possible myocardial infarction) ••• Diast(<65;75)...<85;95))
9	113	0.933	Subsc(<26;28)...<30;32)) & Tric(8...10) ••• Diast(<75;85)...<95;105))
10	91	0.930	Subsc(<18;20), <20;22)) & Chest(non-ischaemic, other) ••• Diast(<65;75)...<85;95))

Fig. 5. Automatically filtered association rules

according to (i) – (iii), only 51 rules remain from the original 158 rules. Several examples are in figure 5.

We can see that there is true rule $BMI\langle 16;21\rangle \Rightarrow_{0.9,30} Diast\langle 65;95\rangle$ which satisfies neither (i) nor (ii) and thus it cannot be considered as a consequence of $BMI \uparrow\uparrow Diast$. This is a reason to study this rule in more detail, because it could be an interesting exception. It should be reported to the domain expert. However, let us emphasize that definition of $Cons(BMI \uparrow\uparrow Diast, Entry, \Rightarrow_{0.9,30})$ in figure 3 was done without a consultation with domain expert.

Additional remaining rules concern attributes *Subsc* and *Diast* in some cases combined with *Chest* and *Tric*. We assume that by a suitable analytical process we can offer a new item of domain knowledge $Subsc \uparrow\uparrow Diast$.

5 Conclusions and Further Work

Here presented approach allows filtering out all rules reasonable considered as consequences of domain knowledge, e.g. the above mentioned $BMI \uparrow\uparrow Diast$. This leads to a remarkable decrease of number of output association rules, so users could concentrate on interpretation of a smaller group of potentially more valuable association rules. An already available implementation has even more filtering features that could be moreover repeated in an iterative way.

Let us emphasize that there are several additional types of mutual influence of attributes [7]. An example is $Education \uparrow\downarrow BMI$ which says that if education increases then BMI decreases. All these types of knowledge can be treated in the above described way [5]. The described transformation of an item of domain knowledge into a set of association rues can be inverted and used to synthesize a new item of domain knowledge (e.g. $Subsc \uparrow\uparrow Diast$).

The whole approach seems to be understandable from the point view of a domain expert. However, a detailed explanation will be useful. This leads to necessity to prepare for each analytical question an analytical report explaining in details all of steps leading to its solution. There are first results in producing similar reports and presenting them at Internet [2].

Our goal is to elaborate the outlined approach into a way of automatized producing analytical reports answering given analytical question. Domain knowledge stored in the LISp-Miner system gives a possibility to automatically generate a

whole system of analytical questions. Successful experiments with running LISp-Miner system on a grid [9] makes possible to accept a challenge to create a system automatically formulating analytical question, getting new knowledge by answering these question and use new knowledge to formulate additional analytical question. Considerations on such a system are in [10].

References

1. Hájek, P., Havránek, T.: Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory. Springer, Heidelberg (1978)
2. Kliegr, T., et al.: Semantic Analytical Reports: A Framework for Post processing data Mining Results. In: Rauch, J., et al. (eds.) Foundations of Intelligent Systems, pp. 88–98. Springer, Heidelberg (2009)
3. Qiang, Y., Xindong, W.: 10 Challenging Problems in Data Mining Research. International Journal of Information Technology & Decision Making 5(4), 597–604 (2006)
4. Rauch, J.: Logic of Association Rules. Applied Intelligence 22, 9–28 (2005)
5. Rauch, J.: Considerations on Logical Calculi for Dealing with Knowledge in Data Mining. In: Ras, Z.W., Dardzinska, A. (eds.) Advances in Data Management. Studies in Computational Intelligence, vol. 223, pp. 177–199. Springer, Heidelberg (2009)
6. Rauch, J., Šimůnek, M.: An Alternative Approach to Mining Association Rules. In: Lin, T.Y., et al. (eds.) Data Mining: Foundations, Methods, and Applications, pp. 219–238. Springer, Heidelberg (2005)
7. Rauch, J., Šimůnek, M.: Dealing with Background Knowledge in the SEWEBAR Project. In: Berendt, B., Mladenič, D., de Gemmis, M., Semeraro, G., Spiliopoulou, M., Stumme, G., Svátek, V., Železný, F., et al. (eds.) Knowledge Discovery Enhanced with Semantic and Social Information. Studies in Computational Intelligence, vol. 220, pp. 89–106. Springer, Heidelberg (2009)
8. Suzuki, E.: Discovering interesting exception rules with rule pair. In: Fuernkranz, J. (ed.) Proceedings of the ECML/PKDD Workshop on Advances in Inductive Rule Learning, pp. 163–178 (2004)
9. Šimůnek, M., Tammisto, T.: Distributed Data-Mining in the LISp-Miner System Using Techila Grid. In: Zavoral, F., Yaghob, J., Pichappan, P., El-Qawasmeh, E. (eds.) NDT 2010. Communications in Computer and Information Science, vol. 87, pp. 15–20. Springer, Heidelberg (2010)
10. Šimůnek, M., Rauch, J.: EverMiner – Towards Fully Automated KDD Process, accepted for publication in Data Mining. In: TECH (2011) ISBN: 978-953-7619-X-X

A Compression-Based Dissimilarity Measure for Multi-task Clustering

Nguyen Huy Thach, Hao Shao, Bin Tong, and Einoshin Suzuki

Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan
{thachnh,shaohao,bintong}@i.kyushu-u.ac.jp,
suzuki@inf.kyushu-u.ac.jp

Abstract. Virtually all existing multi-task learning methods for string data require either domain specific knowledge to extract feature representations or a careful setting of many input parameters. In this work, we propose a feature-free and parameter-light multi-task clustering algorithm for string data. To transfer knowledge between different domains, a novel dictionary-based compression dissimilarity measure is proposed. Experimental results with extensive comparisons demonstrate the generality and the effectiveness of our proposal.

1 Introduction

The intuition behind a multi-task learning (MTL) algorithm [2] is that it tries to retain and reuse knowledge previously learned in one task to solve other learning tasks in different domains. MTL has been applied successfully in diverse domains such as bioinformatics, text mining, natural language processing, image analysis, WIFI location detection, and computer aided design (CAD) [9]. However, virtually all existing MTL methods for string data require either domain-specific knowledge to extract feature representation or a careful setting of many input parameters. For example, the standard document data typically needs to be represented in a special format of the vector space model [15]. The commonly used Gaussian multi-task learning framework in [13] requires a careful setting for the covariance function parameters, a scalar parameter and kernel function parameters. The requirement of domain-specific knowledge in extracting features may limit the applicability of an algorithm to string data sets whose important features are unknown, missing or difficult to be extracted. In addition, for a heavily parameterized multi-task learning algorithm, it is difficult to determine whether the improvement of the performance is from setting the values of the parameters or from using knowledge transferred between different domains.

We focus on building a multi-task learning framework for string data which does not assume a specific feature space and needs only a few parameters. In this work, we propose a *universal* and *parameter-light* Compression-based Multi-task Clustering (*CMC*) framework. It is motivated by the recent successful applications of methods based on Kolmogorov complexity [12] on various data types including music, time series, images, natural language and genomic data ([16] and references therein). However these methods are only defined for traditional single data mining tasks. To handle this issue,

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 123–132, 2011.

we create a Learnable Dictionary-based Compression Dissimilarity Measure (*LDCDM*) that allows to transfer knowledge between learning tasks effectively. In our *CMC* framework, for each domain, it requires only a setting of one parameter, which is the number of clusters, and it does not require any specific set of features.

2 Related Works

There are some works based on Kolmogorov theory to calculate the relatedness between tasks in supervised transfer learning. [11] attempts to exploit the probability theory of Bayesian learning to measure the amount of information that one task contains about another. [6] creates a measure based on a compression algorithm and the probably approximately correct (PAC) learning to group similar tasks together. Mahmud approximates Kolmogorov complexity of a decision tree by the number of its nodes and the conditional complexity of two decision trees based on the maximum number of their overlapping nodes [10]. However, one thing all these works have in common is that for different learning algorithms, there are different strategies to measure the relationship between tasks. For example, the method proposed for a Bayesian learning algorithm must be redefined if it is applied to induction of decision trees. In this work, we propose a distance measure for string data that does not assume any specific learning algorithms.

In our setting, we assume that labeled data are unavailable. The learning problem in this scenario is extremely difficult typically due to the fact that the data in the input domains have different distributions. So far, there are some MTL research works in this setting. Indrajit et al. propose a cross-guided clustering algorithm for text mining, where a similarity measure based on pivot words across domains is introduced to discover and measure hidden relationships between the source and target domains [4]. Gu and Zhou propose a cross-task clustering framework, which aims at learning a subspace shared by all tasks [5]. Recently, Zhang et al. propose a Bregman-based clustering algorithm to solve the multi-task problem [18]. However, different from our work, these works are feature-based and they need at least three parameters in their algorithms.

3 Preliminaries

The Kolmogorov complexities $K(x)$, $K(x|y)$ and $K(xy)$ of a finite string x, x given y and the concatenation xy of x and y are the lengths of the shortest programs to generate x, x when y is given as an input and xy on a universal computer such as the Turing machine, respectively. In [12], the authors define an information distance, $d_k(x, y)$, between two strings x and y using the Kolmogorov complexities as:

$$d_k(x, y) = \frac{K(x|y) + K(y|x)}{K(xy)} \tag{1}$$

The d_k measure has been shown to be a lower bound and optimal of all measures of information content in [12]. Unfortunately, the Kolmogorov complexity is uncomputable in general and thus one must use its approximation. The main idea for the approximation is to respectively substitute $K(x|y)$, $K(y|x)$ and $K(xy)$ with $Comp(x|y)$,

$Comp(y|x)$ and $Comp(xy)$, where $Comp(xy)$ is the compressed size of xy and $Comp(x|y)$ is the compressed size of x achieved by first training the compressor on y, and then compressing x. The d_k measure is then approximated by d_c [12] as follows:

$$d_c(x, y) = \frac{Comp(x|y) + Comp(y|x)}{Comp(xy)} \tag{2}$$

The better the compression algorithm is, the better the approximation of d_c for d_k is.

To calculate $Comp(x|y)$, we need a compression algorithm that allows us to compress x given y. This necessity has led us to consider *dictionary-based* compression algorithms [17] where one instance can be used as an auxiliary input to compress another instance. The compressor first builds a dictionary on y, then x is scanned to look for its repeated information stored in the dictionary, and finally the repeated information in x is replaced by a much shorter index to reduce the size of x. If the information of x is not contained in the dictionary, the dictionary will be updated by the new information. The more closely related x and y are, the fewer number of times the dictionary is updated or the smaller $Comp(x|y)$ is. $Comp(x)$ can be considered as a special case of $Comp(x|y)$ where y is an empty string. In this work, we choose Lempel-Ziv-Welch (LZW), a dictionary-based compression [17], and use it in our framework because it is a lossless, linear and universal compressor with no specific assumptions about the input data and the fact that its implementations tend to be highly optimized. The detail of LZW algorithm is omitted here for brevity, so interested readers are referred to [17] or `http://marknelson.us/1989/10/01/lzw-data-compression/` for LZW implementation.

4 Compression-Based Multi-task Clustering

4.1 Intuition Behind Our Method

The intuition behind our method is illustrated in Fig. 1a, where a clustering task is being considered and two clusters with two centroids $\mathbf{c}_1$ and $\mathbf{c}_2$ are obtained. In this example, instance $\mathbf{x}_1$ may be naturally included into cluster $\mathbf{c}_1$ but instance $\mathbf{x}_2$ may not be clustered correctly if we use the dissimilarity measure d_c in Eq. (2) because $d_c(\mathbf{x}_2, \mathbf{c}_1) = \mathrm{d_c}(\mathbf{x}_2, \mathbf{c}_2)$. However, if we consider $\mathbf{x}_1$, a neighbor instance of $\mathbf{x}_2$, in

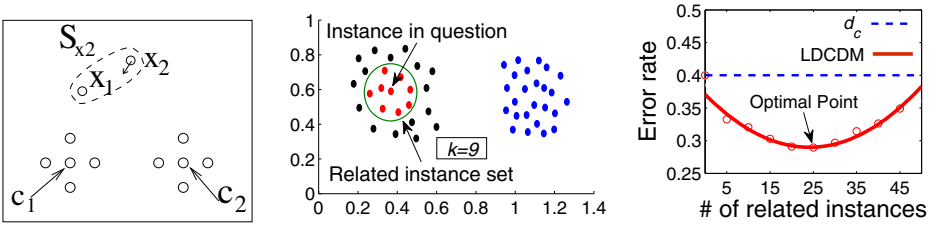


Fig. 1. (*left*) An illustrative example, (*center*) an example of selecting related instance by kNN algorithm, (*right*) the error rates of d_c and $LDCDM$ on increasing numbers of related instances

measuring the dissimilarity, we may obtain useful information to help cluster $\mathbf{x}_2$. We therefore decide to extend the distance d_c in Eq. (2) further by learning more information from neighbor instances and propose a Learnable Dictionary-based Compression Dissimilarity Measure (*LDCDM)* as follows:

$$LDCDM(x,y) = \frac{Comp(x|S_y) + Comp(y|S_x)}{Comp(xy)} \tag{3}$$

where S_x and S_y are neighbor instance sets of x and y, respectively. These neighbor instances are regarded as related instance sets and how to build them are explained in the next paragraph. In the example of Fig. 1a, $S_{x_2} = \{\mathbf{x}_1, \mathbf{x}_2\}$, the dictionary built on S_{x_2} contains more information about $\mathbf{c}_1$ than $\mathbf{x}_2$ does, so $Comp(\mathbf{c}_1|\mathrm{S}_{\mathrm{x}_2}) < \mathrm{Comp}(\mathbf{c}_1|\mathbf{x}_2)$ or $LDCDM(\mathbf{x}_2, \mathbf{c}_1) < \mathrm{LDCDM}(\mathbf{x}_2, \mathbf{c}_2)$. Therefore, $\mathbf{x}_2$ is reasonably included into cluster $\mathbf{c}_1$ instead of cluster $\mathbf{c}_2$.

We explain how to build related instance sets, S_x and S_y, in this sub-section and the next sub-section. Recall that, the smaller $Comp(x|y)$ is, the more closely related x and y are, and $Comp(x)$ is the compressed size of x without any prior information of other instances. Therefore, we define the *relatedness degree*, $\triangle(x|y)$, of x and y as:

$$\triangle\,(x|y) = Comp(x) - Comp(x|y) \tag{4}$$

To investigate the effectiveness of $LDCDM$ compared to d_c, both $LDCDM$ and d_c are used in the *k*-means algorithm to cluster 20 Newsgroups data set[1] (more details of this document data set are given in Section 5.2). The relatedness degree $\triangle$ is used to build S_x and S_y, where S_x and S_y contain the k nearest neighbor instances of x and y based on values of $\triangle$, respectively. Fig. 1b shows an example of selecting related instances with $k = 9$. For clarity of exposition, we only select a subset of the 20 Newsgroups data, for instance, in this case, we select $n = 25$ documents from each of two classes, *comp.* and *rec.*, in domain 1 (see Table 2 for more information). Fig. 1c shows the results of d_c and $LDCDM$ in different numbers of related instances being selected. The class labels are used as a ground truth, so it is possible to evaluate a clustering result with its error rate, where an error rate is the number of wrongly clustered instances divided by the total number of the input instances. As we can see, the error rate of $LDCDM$ gradually decreases as new related instances are added. $LDCDM$ achieves the optimal result when the number *k* of related instances is $k = n = 25$. This experiment shows that by using related instances, the performance of a compression-based clustering method can be improved. However, we need to provide the value of k that may require some domain knowledge. In the next section, we describe a general multi-task clustering framework which does not require setting the value of k.

4.2 *CMC* Framework with *LDCDM* Measure

Suppose we are given *m* clustering tasks $T_1, T_2, \ldots, T_m$, and T_i is accompanied with a domain D_i. Each domain D_i consists of two components: a feature space $\mathcal{X}_i$ and a marginal probability distribution $P(X_i)$ where $X_i = \{\mathbf{x}_1^{\mathrm{i}}, \mathbf{x}_2^{\mathrm{i}}, \ldots, \mathbf{x}_{\mathrm{n_i}}^{\mathrm{i}}\} \in \mathcal{X}_{\mathrm{i}}$.

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

Algorithm 1. Compression-based Multi-task Clustering (CMC) Algorithm

Input: + maximum number of iterations *MAX_ITER*
+ m tasks, $T = \{T_1, T_2, \ldots, T_m\}$, each T_i has one data set $X_i = \{x_1^i, x_2^i, \ldots, x_n^i\}$.
+ Vector $K = \{k_1, k_2, \ldots, k_m\}$ where k_i is the desired number of clusters in task T_i.
Output: k_i clusters of each task T_i $(i = 1, 2, \ldots, m)$.

1: **Initialization:** Set $t = 0$ and apply the k-means algorithm to cluster each X_i $(i = 1, 2, \ldots, m)$ into k_i clusters, $C^i = \{C_1^i, C_2^i, \ldots, C_{k_i}^i\}$ by using d_c in Eq. (2).
2: **repeat**
3: **for** $i = 1$ to m **do**
4: **for** each x_j^i in X_i **do**
5: $S_{x_j^i} \leftarrow findRelatedInstances(x_j^i)$
6: **for** $i = 1$ to m **do**
7: $D_i \leftarrow calculateDistanceMatrix(X_i)$ /* Calculate distance matrix of X_i */
8: Apply the k-means algorithm on D_i to update clusters in C^i.
9: $t = t + 1$
10: **until** all $\{C^i\}_{i=1}^m$ do not change or $t \geq MAX\_ITER$

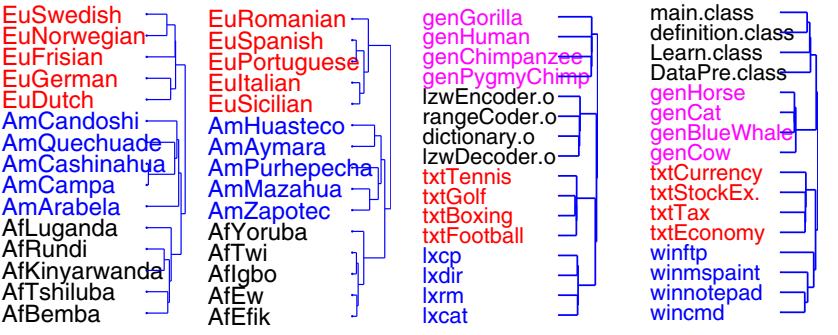
The clustering algorithm partitions the data X_i into k_i non-overlapping clusters $C^i = \{C_1^i, C_2^i, \ldots, C_{k_i}^i\}$. The intuition behind our approach is that although we cannot reuse all data of one domain in other domains, there are certain parts that can still be reused.

The motivating example in the previous section illustrates the effectiveness of using related instances in a compression-based dissimilarity measure for a single task learning. However, applying this method directly to multi-task learning is not straightforward. In multi-task learning, the distributions of different domains may not be the same, therefore a value of k which is suitable for one domain may not be suitable for other domains. A reasonable strategy is to seek a different value of k for each domain. However, such an approach may require some domain knowledge and violates our goal of using as few parameters as possible. Since instances in the same cluster of a clustering result are usually related to each other, we are motivated to propose a framework where multiple clustering operations are performed simultaneously and the result of one cluster operation is used to help find related instances in the other clustering operations.

Using the above analysis and the definition of *LDCDM*, our *Compression-based Multi-task Clustering* (*CMC*) framework is given in Algorithm 1. *CMC* includes an *initialization* step and a loop which consists of 2 subroutines: *findRelatedInstances* and *calculateDistanceMatrix*. In the *initialization* step, the dissimilarity measure d_c in Eq. (2) is employed to calculate distance matrices and the *k-means* algorithm is used to cluster each data set X_i into k_i clusters, $C^i = \{C_1^i, C_2^i, \ldots, C_{k_i}^i\}$ (line 1). Then, for each instance, the *findRelatedInstances* procedure is used to find related instances (line 3-5) as follows. We compute the distance between one instance x and one cluster C_i by the distance between x and the centroid, $Centroid(C_i)$, of C_i:

$$d_{instance\_cluster}(x, C_i) = d_c(x, Centroid(C_i)) \tag{5}$$

Each instance x has one closest cluster within its domain and (m - 1) closest clusters across domains. The closest cluster of one instance is defined as the cluster having the minimum distance from its centroid to the instance. The set of instances in the



(a) Languages 1 (b) Languages 2 (c) Heterogeneous 1 (d) Heterogeneous 2

Fig. 2. Intuitive results in *Language* and *Heterogeneous* data

closest clusters is regarded as a candidate set of related instances. To filter out unrelated instances from the candidate set, the *relatedness degree* in Eq. (4) is employed (line 4 of *findRelatedInstances* procedure). Finally, the procedure returns the related instance set S_x.

Procedure $findRelatedInstances(x)$

1: $S_x \leftarrow \emptyset$ /* Initialized to an empty set */
2: **for** $i = 1$ to m **do**
3: $\quad S_x \leftarrow S_x \cup \arg\underset{C_j^i}{min}(\{d_{instance\_cluster}(x, C_j^i)\}_{j=1}^{k_i})$
4: $S_x \leftarrow \{y | y \in S_x$ and $\triangle(x|y) > 0\}$ /* filter out unrelated instances */
5: **return** S_x

In line 7 of Algorithm 1, our *LDCDM* measure is used in the *calculateDistanceMatrix* procedure to compute the dissimilarity matrix D_i of the $i-$th domain. Then the dissimilarity matrix is used to re-calculate k_i new clusters. This process is iterated until the clusters in all $\{C^i\}_{i=1}^m$ do not change.

Procedure $calculateDistanceMatrix(X_i)$

1: **for** $j = 1$ to n_i **do**
2: $\quad$ **for** $l = 1$ to n_i **do**
3: $\quad\quad D_i(j, l) \leftarrow LDCDM(x_j^i, x_l^i, S_{x_j^i}, S_{x_l^i})$
4: **return** D_i

*

5 Experimental Results

5.1 Results on Language and Heterogeneous Data Sets

We begin this section by a comprehensive set of experiments where the outcome can be checked directly by human. The test consists of experiments on two data sets:

Table 1. Experimental Results of Language and Heterogeneous Data Sets

Domain	All		NCD		CMC	
	Acc	NMI	Acc	NMI	Acc	NMI
Language 1	0.60	0.58	1.0	1.0	1.0	1.0
Language 2	0.73	0.62	1.0	1.0	1.0	1.0
Heterogeneous 1	0.88	0.77	0.81	0.78	1.0	1.0
Heterogeneous 2	0.81	0.70	0.81	0.71	1.0	1.0

a *linguistic corpus* and a *natural heterogeneous* data, both of which have been previously used in [16]. The *linguistic corpus* is a collection of "The Universal Declaration of Human Rights" [2] (UDHR)" in 30 different languages which are from 3 main language families[3] under which are 6 language groups: *Europe* (*Italic* and *Germanic* groups), *America* (*Mexico* and *Peru* groups) and *Africa* (*Bantu* and *Kwa* groups). We define the domains based on the *main* language families while the data are split based on the *groups*. Therefore, we have two different but related domains: domain 1 consists of languages from *Italic*, *Mexico* and *Bantu* while domain 2 consists of languages from *Germanic*, *Peru* and *Kwa*. The *natural heterogeneous* data set includes 32 files from 4 different file types divided into 8 sub-categories as follows: (a) 8 gene sequence files (from 2 groups: *Primates* and *Ferungulates*), downloaded from GenBank[4]; (b) 8 compiled files (from C++ and Java) of our CMC program; (c) 8 text files (from 2 topics: *Sports* and *Finance*), downloaded from Wikipedia[5]; (d) 8 executive files, copied directly from our Ubuntu 9.10 and Windows XP systems. The heterogeneous data is also split into two domains as the manner in the language data set.

To illustrate the effectiveness of the multi-task framework, for each data set, we have combined all the data from both domains and applied the proposed dissimilarity measure on the combined data. We only evaluate the experimental result of the data in each single domain in accordance with the setting of multi-task learning. The result of one widely used Kolmogorov-based algorithms NCD [16] is also included. Two metrics *Clustering Accuracy* (*Acc*) and *Normalized Mutual Information* (*NMI*) [1] are employed to evaluate the clustering performance. The higher the values on *Acc* and *NMI* are, the better are the clustering results. The results of these methods are shown in Table 1 and in CMC homepage[6], where "All" refers to the results of combining data. From the table, we can see that CMC always gives the best performances. Results of combining data indicate that simply combining data does not help to improve the clustering performance because the distributions of combined data are different. Besides, NCD performs as well as CMC in language data but worse than CMC in heterogeneous data. These results demonstrate the effectiveness of our framework for these problems which consist of various kinds of data.

[2] http://www.un.org/en/documents/udhr/
[3] http://www.freelang.net/families/
[4] http://www.ncbi.nlm.nih.gov/genbank/
[5] http://www.wikipedia.org/
[6] http://tlas.i.kyushu-u.ac.jp/~suzuki/ISMIS2011/ISMIS2011.html

Table 2. Description of 20 Newsgroups Data Set. We hide class labels in the experiments and try to recover them by clustering.

Domain	Class label				total # doc.
	comp.	rec.	sci.	talk.	
1	graphics	auto	crypt	politics.guns	800
2	os.ms-win.misc	motorcycle	electronics	politics.mideast	800
3	sys.ibm.pc.hw	sport.baseball	med	politics.misc	800
4	sys.mac.hw	sport.hockey	space	religion.misc	800

5.2 Results on Document Data

The 20 Newsgroups data set is widely used in the multi-task learning community, which is a collection of approximately 20,000 newsgroup documents, partitioned evenly across 6 root categories under which are 20 subcategories. We define the data class labels based on the root categories, namely, *comp*, *rec*, *sci* and *talk*. Then the data are split based on sub-categories, which ensures that the two data domains contain data in different but related distributions. The detailed constitution of the 20 Newsgroups data is summarized in Table 2. For preprocessing document data, we applied the same process as in [14]: we removed the header lines and the stop words and selected the top 2000 words by mutual information. In each domain, we randomly selected 200 documents, so each task has 800 documents. For the competitive methods, in order to satisfy their requirements of the data representation, we represent each document as a vector in the vector space model [15] with the number of attributes being 2000 (equal to the number of words). In contrast, our method uses the preprocessed text data directly.

Our method is compared to a large collection of both single and multi-task clustering algorithms: NCD [16] and CDM [8]; 4 Bernoulli model-based methods [15]: *kberns*, *skberns*, *mixberns* and *bkberns*; 1 state-of-the-art graph-based approach, CLUTO [7]; 1 co-clustering algorithm [3] whose idea is similar to multi-task learning with two clustering operations on columns and rows of the data matrix performing simultaneously. The experimental results of 2 multi-task clustering algorithms: Learning the Shared Subspace for Multi-Task Clustering (LSSMTC) [5] and Multitask Bregman Clustering (MBC) [18] are also included. The cross-guided clustering algorithm [4] is not compared because, unlike ours, it requires prior knowledge on some pivot words. We set the number of clusters equal to the number of classes for all the clustering algorithms. For setting parameter values, we follow the settings recommended by the competitive methods, which are considered to be optimal. For each algorithm, we repeat clustering 10 times, then we report the average and the standard deviation of *Acc* and *NMI* values.

The experimental results of CMC and competitive methods are shown in Table 3. In general, in most cases, CMC outperforms other methods. We can see that the clustering results of CMC improve over the clustering results of CDM and NCD, which are single compression-based algorithms. This improvement owes to the proposed dissimilarity measure which exploits the relation among the tasks of CMC. The results of CLUTO and co-clustering are also noticeable, e.g., CLUTO, in task 1, exhibits the

Table 3. Experimental Results of 20 Newsgroups Data Set

Method	Task 1		Task 2		Task 3		Task 4	
	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI
CDM	.72±.07	.63±.05	.73±.07	.64±.05	.73±.07	.63±.06	.72±.08	.62±.06
NCD	.76±.12	.65±.14	.69±.14	.52±.12	.69±.14	.51±.17	.67±.09	.52±.09
kberns	.53±.07	.36±.08	.53±.07	.39±.09	.51±.08	.33±.05	.59±.09	.41±.06
skberns	.55±.04	.36±.05	.51±.07	.38±.07	.52±.08	.36±.09	.50±.05	.43±.06
mixberns	.50±.08	.33±.06	.55±.09	.40±.10	.50±.07	.34±.05	.57±.07	.41±.08
bkberns	.61±.12	.46±.11	.63±.04	.39±.05	.62±.08	.46±.07	.65±.08	.52±.09
CLUTO	**.81±.06**	.57±.05	.77±.02	.54±.03	.75±.01	.51±.02	.80±.05	.62±.04
co-clustering	.69±.10	.48±.04	.72±.04	.51±.03	.59±.14	.46±.11	.68±.09	.53±.04
MBC	.61±.11	.40±.10	.63±.06	.47±.05	.61±.07	.38±.06	.62±.07	.41±.06
LSSMTC	.65±.06	.42±.05	.60±.02	.44±.02	.64±.09	.43±.09	.66±.07	.46±.05
CMC	**.81±.07**	**.71±.06**	**.81±.03**	**.70±.04**	**.83±.08**	**.73±.06**	**.82±.08**	**.72±.09**

best performance in terms of *Acc*. The Bernoulli-based algorithms underperform other methods because, in these methods, a document is represented as a binary vector while the numbers of word occurrences are not considered [15].

To illustrate the weakness of feature-based and parameter-laden algorithms, we examined MBC and LSSMTC on 20 Newsgroups data with different settings of feature extraction and parameters. When the number of documents on each task is 200 as in Table 2, we found that MBC obtained the optimal result as shown in Table 3 once the number of features is equal to 43,000. However, when we add more 50 documents to each task, MBC could not obtain the optimal performance with 43,000 features. For LSSMTC, we also found that LSSMTC could obtain its optimal performance if we tune its parameters. However, once the parameters are tuned on the new data set, LSSMTC could not converge to the optimal solution on the old data set. On the other hand, our proposal is feature-free and parameter-light, so it does not encounter the same problems. The experimental results on this section illustrate the main point of this paper: with a feature-free and parameter-light algorithm, we can avoid the overfitting problem.

Because CMC is an iterative algorithm, it is important to evaluate the convergence property. In this paper, we show the convergence of CMC empirically and we are going to prove it in the next version. Figure 3 shows the error rate curves as functions of the number iterations of CMC on all data sets used in this paper. From this figure, we can see that CMC practically converges after 17 iterations. Note that, in our experiments, we set the maximum number of iterations, $MAX\_ITER = 30$.

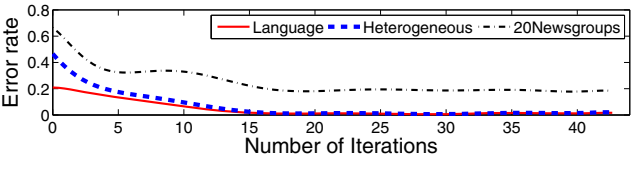


Fig. 3. Convergence curves of CMC on *Language*, *Heterogeneous* and *20 Newsgroups* data sets

6 Conclusions

In this paper, we introduced, for the first time, a universal and parameter-light multi-task clustering framework for string data. Our proposal can be applied to a wide range of data types and it only requires the number of clusters for each domain as the input parameter. A compression-based dissimilarity measure is proposed to utilize related instances within and across domains in the setting of multi-task learning to improve the clustering performance. Experimental results on *linguistic corpus*, *heterogeneous* and *document* data sets demonstrate the generality and effectiveness of our proposal. The experiments also show that our universal and parameter-light algorithm almost always outperforms other methods including parameter-laden and feature-based algorithms even in domains which they are specifically designed for.

References

1. Cai, D., He, X., Wu, X., Han, J.: Non-negative Matrix Factorization on Manifold. In: ICDM, pp. 63–72 (2008)
2. Caruana, R.: Multitask Learning. Machine Learning 28, 41–75 (1997)
3. Dhillon, I.S.: Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning. In: KDD, pp. 269–274 (2001)
4. Indrajit, B., et al.: Cross-Guided Clustering: Transfer of Relevant Supervision across Domains for Improved Clustering. In: ICDM, pp. 41–50 (2009)
5. Gu, Q., Zhou, J.: Learning the Shared Subspace for Multi-task Clustering and Transductive Transfer Classification. In: ICDM, pp. 159–168 (2009)
6. Juba, B.: Estimating Relatedness via Data Compression. In: ICML, pp. 441–448 (2006)
7. Karypis, G., Kumar, V.: A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM J. Sci. Comput. 20, 359–392 (1998)
8. Keogh, E., Lonardi, S., Ratanamahatana, C.A.: Towards Parameter-free Data Mining. In: KDD, pp. 206–215 (2004)
9. Liu, Q., Liao, X., Carin, H.L., Stack, J.R., Carin, L.: Semisupervised Multitask Learning. IEEE Trans. on PAMI 31, 1074–1086 (2009)
10. Mahmud, M.M.H.: On Universal Transfer Learning. In: Hutter, M., Servedio, R.A., Takimoto, E. (eds.) ALT 2007. LNCS (LNAI), vol. 4754, pp. 135–149. Springer, Heidelberg (2007)
11. Mahmud, M.M.H., Ray, S.R.: Transfer Learning Using Kolmogorov Complexity: Basic Theory and Empirical Evaluations. In: NIPS, pp. 985–992 (2008)
12. Ming, L., Paul, V.: An Introduction to Kolmogorov Complexity and its Applications, 2nd edn. Springer, New York (1997)
13. Schwaighofer, A., Tresp, V., Yu, K.: Learning Gaussian Process Kernels via Hierarchical Bayes. In: NIPS, pp. 1209–1216 (2004)
14. Slonim, N., Tishby, N.: Document Clustering Using Word Clusters via the Information Bottleneck Method. In: SIGIR, pp. 208–215 (2000)
15. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In: KDD Workshop on Text Mining, pp. 25–36 (2000)
16. Vitanyi, P.M.B., Balbach, F.J., Cilibrasi, R., Li, M.: Normalized Information Distance. In: CoRR, abs/0809.2553 (2008)
17. Welch, T.: A Technique for High-Performance Data Compression. Computer 17, 8–19 (1984)
18. Zhang, J., Zhang, C.: Multitask Bregman Clustering. In: AAAI (2010)

Data Mining in Meningoencephalitis: The Starting Point of Discovery Challenge

Shusaku Tsumoto[1] and Katsuhiko Takabayashi[2]

[1] Department of Medical Informatics, Faculty of Medicine, Shimane University, 89-1 Enya-cho Izumo 693-8501 Japan
tsumoto@computer.org
[2] Division of Medical Informatics, Chiba University Hospital, 1-8-1 Inohana, Chiba 260 Japan
takab@ho.chiba-u.ac.jp

Abstract. The main difference between conventional data analysis and KDD (Knowledge Discovery and Data mining) is that the latter approaches support discovery of knowledge in databases whereas the former ones focus on extraction of accurate knowledge from databases. Therefore, for application of KDD methods, domain experts' interpretation of induced results is crucial. However, conventional approaches do not focus on this issue clearly. In this paper, 11 KDD methods are compared by using a common medical database and the induced results are interpreted by a medical expert, which enables us to characterize KDD methods more concretely and to show the importance of interaction between KDD researchers and domain experts.

1 Introduction

Statistical pattern recognition methods and empirical learning method [9] have been developed in order to acquire accurate knowledge which is similar to that of domain experts. On the other hand, knowledge discovery in databases and data mining (KDD) [4,8] has a different goal, to extract knowledge which is not always expected by domain experts, which will lead to a new discovery in applied domain. For this purpose, the evaluation of predictive accuracy [9] is not enough and domain experts'interpretation of induced results is crucial for discovery. However, conventional approaches do not focus on this issue clearly. In this paper, eleven rule induction methods were compared by using a common medical database on meningoencepahlitis. The induced results were interpreted by a medical expert, which showed us that rule induction methods generated unexpected results, whereas decision tree methods and statistical methods acquired knowledge corresponding to medical experts. These results enable us to characterize KDD methods more concretely and to show the importance of interaction between KDD researchers and domain experts.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 133–139, 2011.

2 Database on Meningoencephalitis

2.1 Information about Data

The common datasets collect the data of patients who suffered from meningitis and were admitted to the department of emergency and neurology in several hospitals. The author worked as a domain expert for these hospitals and collecting those data from the past patient records (1979 to 1989) and the cases in which the author made a diagnosis (1990 to 1993).

The database consists of 121 cases and all the data are described by 38 attributes, including present and past history, laboratory examinations, final diagnosis, therapy, clinical courses and final status after the therapy, whose information is summarized in Table 1 and 2.[1]

Important issues for analyzing this data are: to find factors important for diagnosis (DIAG and DIAG2), ones for detection of bacteria or virus(CULT_FIND and CULTURE) and ones for predicting prognosis(C_COURSE and COURSE). Also, associative relationships between observations and examinations are very interesting issues because some laboratory examinations are invasive to which complications should be taken account.

Table 1. Attributes in Dataset

Category		#Attributes
Present History	Numerical and Categorical	7
Physical Examination	Numerical and Categorical	8
Laboratory Examination	Numerical	11
Diagnosis	Categorical	2
Therapy	Categorical	2
Clinical Course	Categorical	4
Final Status	Categorical	2
Risk Factor	Categorical	2
Total:		38

2.2 Statistical Analysis and Experts' Prejudice

The author analyzed the subset of this database(99 cases), which was collected until 1989 by using the t-test and χ^2-test and reported to the conference on acute medicine in Japan[7]. Before domain experts usually apply statistical methods to a database, they remove some attributes from a dataset according to their knowledge from a textbook[1] or the literature[3,5]. In the case of the analysis above, age and sex are removed from a dataset since information about these attributes is not in such a textbook.

Concerning numerical data, body temperature, Kernig sign, CRP, ESR, and CSF cell count had a statistical significance between bacteria and virus meningitis. As to categorical data, loss of consciousness and the finding in CT had a

[1] Except for one attribute, all the attributes do not have any missing values.

Table 2. Information about Attributes

I. Personal Information	
1. AGE:	Age
2. SEX:	Sex
II.Diagnosis	
3. DIAG:	Diagnosis described in a database
4. Diag2:	Grouped Attribute of DIAG (Grouped)
III. Present History	
5. COLD:	Since when the patient has symptoms like common cold. (0:negative)
6. HEADACHE:	Since when he/she has a headache. (0:no headache)
7. FEVER:	Since when he/she has a fever. (0:no fever)
8. NAUSEA:	when nausea starts (0:no nausea)
9. LOC:	when loss of consciousness starts (0: no LOC)
10. SEIZURE:	when convulsion or epilepsy is observed (0: no)
11. ONSET:	{ACUTE,SUBACUTE,CHRONIC,RECURR: recurrent }
IV. Physical Examinations at Admission	
12. BT:	Body Temprature
13. STIFF:	Neck Stiffness
14. KERNIG:	Kernig sign
15. LASEGUE:	Lasegue sign
16. GCS:	Glasgow Coma Scale (Min: 3 (comatose), Max: 15(normal))
17. LOC_DAT:	loss of consciousness (-: negativeA +: positive) (Grouped)
18. FOCAL:	Focal sign (-: negativeA +: positive) (Grouped)
V. Laboratory Examinations at Admission	
19. WBC:	White Blood Cell Count
20. CRP:	C-Reactive Protein
21. ESR:	Blood Sedimentation Test
22. CT_FIND:	CT Findings (Grouped)
23. EEG_WAVE:	Electroencephalography(EEG) Wave Findings (Grouped)
24. EEG_FOCUS:	Focal Sign in EEG
25. CSF_CELL:	Cell Count in Cerebulospinal Fluid
26. Cell_Poly:	Cell Count (Polynuclear cell) in CSF
27. Cell_Mono:	Cell Count (Mononuclear cell) in CSF
28. CSF_PRO:	Protein in CSF
29. CSF_GLU:	Glucose in CSF
30. CULT_FIND:	Whether bacteria or virus is specified or not. (T: found, F: not found) (Grouped)
31. CULTURE:	The name of Bacteria or Virus (-: not found)
VI. Therapy and Clinical Courses	
32. THERAPY2:	Therapy Chosen by Neurologists
33. CSF_CELL3:	Cell Count CSF 3 days after the treatment ((Missing values included))
34. CSF_CELL7:	Cell Cound of CSF 7 days after the treatment
35. C_COURSE:	Clinical Course at discharge
36. COURSE(Grouped):	Grouped attribute of C_COURSE (n:negative, p:positive)
VII. Risk Factor	
37. RISK:	Risk Factor
38. RISK(Grouped):	Grouped attribute of RISK (n:negative, p:positive)

Attributes with a label "*" are used as decision attributes (target concepts).

statistical significance between two groups. Also, the analysis suggested that the finding in CT should be an important factor for the final status of meningitis.

However, all these results were expected by medical experts, which means that the author did not discover new knowledge.

3 Preliminary Results

A rule induction method proposed by Tsumoto and Ziarko [13] generated 67 for viral meningitis and 95 for bacterial meningitis, which included the following rules unexpected by domain experts as shown below.[2]

```
1. [WBC < 12000]^[Sex=Female]^[CSF_CELL < 1000] -> Viral
                                                (Accuracy:0.97, Coverage:0.55)
2. [Age > 40]^[WBC > 8000] -> Bacterial         (Accuracy:0.80, Coverage:0.58)
3. [WBC > 8000]^[Sex=Male] -> Bacterial         (Accuracy:0.78, Coverage:0.58)
4. [Sex=Male]^[CSF_CELL>1000]-> Bacterial (Accuracy:0.77, Coverage:0.73)
5. [Risk_Factor=n]->Viral                       (Accuracy:0.78, Coverage:0.96)
6. [Risk_Factor=n]^[Age <40] -> Viral           (Accuracy:0.84, Coverage:0.65)
7. [Risk_Factor=n]^[Sex=Female] -> Viral        (Accuracy:0.94, Coverage:0.60)
```

These results show that sex, age and risk factor are very important for diagnosis, which has not been examined fully in the literature[3,5].

From these results, the author examined relations among sex, age, risk_factor and diagnosis and discovered the interesting relations among them:

(1) The number of examples satisfying [Sex=Male] is equal to 63, and 16 of 63 cases have a risk factor: 3 cases of DM, 3 cases of LC and 7 cases of sinusitis.
(2) The number of examples satisfying [Age≥40] is equal to 41, and 12 of 41 cases have a risk factor: 4 cases of DM, 2 cases of LC and 4 cases of sinusitis.

DM an LC are well known diseases in which the immune function of patients will become very low. Also, sinusitis has been pointed out to be a risk factor for bacterial meningitis[1]. It is also notable that male suffer from DM and LC more than female.

In this way, reexamination of databases according to the induced rules discovered several important knowledge about meningitis.

4 Rule Induction Methods

Inspired by preliminary results shown in Section 3, we organized a discovery contest as 42nd KBS research meeting in Japan AI Society. In this section, all the methods applied to this method are summarized. For precise information about each method, please refer to the proceedings[14].

[2] Rules from 5 to 7 are newly induced by introducing attributes on risk factors.

4.1 SIBLE:Interactive Evolutionary Computation

SIBLE Procedure. Terano and Inada applied SIBLE(Simulated Breeding and Inductive Learning)[11] to the common dataset. This system is a novel tool to mine efficient decision rules from data by using both interactive evolutionary computation(IEC) and inductive learning techniques. The basic ideas are that IEC or simulated breeding is used to get the effective features from data and inductive learning is used to acquire simple decision rules from the subset of the applied data. In this contest, SIBLE was used in the following ways: (1) repeat an apply-and-evaluate loop of C4.5 by a person with medical knowledge loop of C4.5 by a person with medical knowledge to assess the performance of the program; and then (2) apply our GA-based feature selection method with human-in-a-loop interactive manner.

Results. Concerning diagnosis, SIBLE induced several rules by using original C4.5, which correspond to medial experts' knowledge with good performance. Some rules are:

```
1. [Cell_Poly > 220] -> Bacterial (95.9%)
2. [Cell_Poly <=220] ^ [Cell_Mono >12] -> Virus (97.3%).
```

However, these induced rules are not generated by SIBLE method. Then, Terano and Inada applied SIBLE method to induction of rules with respect to prognosis, which is a more difficult problem. This analysis discovered two interesting rules:

```
1. [LOC >6] -> [Prognosis=dead] (50.0%)
2. [LOC <=2] -> [Prognosis=good] (83.6%),
```

which shows that if a patient with loss of consciousness(LOC) came to the hospital within two days after LOC was observed, then his/her prognosis is good.

4.2 GDT-RS

Zhong, J. et al. applied GDT-RS[15], which combines generalization distribution table (GDT) and rough set method(RS) to discover *if-then* rules. GDT provides a probabilistic basis for evaluating the strength of a rule and RS is used to find minimal relative reducts from the set of rules with larger strengths. The strength of a rule represents the uncertainty of the rule, which is influenced by both unseen instances and noises. GDT-RS discovered interesting results for the prognosis problem:

```
1.[STIFF=2]^[KERNIG=0]^[FOCAL=+]^[CSF_PRO>=100]^[RISK=negative]
    =>[Prognosis = not good]  (Strength=2079),
2.[LOC=0]^[STIFF=2]^[EEG_FOCUS=-]^[CSF_PRO>=100]^[CSF_GLU<56]
    =>[Prognosis = not good]  (Strength=1512),
```

where STIFF(neck stiffness), CSF_PRO and CSF_GLU are selected as important factors for prognosis.

4.3 Association Rules Analysis with Discretization

Tsukada, et al. used basket analysis, which induces association rules. First, they adopted MDL principle[6] and AIC[2] to discretize numerical attributes and then induced rules with discretized numerical and categorical attributes [12]. Although MDL principal has been reported to be better than AIC for model selection, experimental results show that AIC discretization is much more useful for medical experts than MDLP.

Basket analysis with AIC discovered the following interesting rules:

```
1.[HEADACHE:[3,63]]^[CELL_Poly:[0,220]]^[EEG_FOCUS:-]^[LOC_DAT:-]
              => Viral  (support=33.6%, confidential=90.4%),
2.[EEG_FOCUS:-]^[SEX:F] => [CRP:[0.0,4.0]]
                          (support=26.4%, confidential=92.5%),
3.[CSF_Poly:[0,220]] => [CRP:[0.0,4.0]]
                           (support=72.9%, confidential=95.3%).
```

The most interesting rule is second one, which suggests that EEG_FOCUS should be related with the value of CRP.

4.4 Exception Rule Discovery

Suzuki applied exception rule discovery[10] based on a hypothesis-driven approach to the common medical dataset. This method searches for rule pairs which consist of a commonsense rule and an exception rule by using conditional probabilities as indices. Interesting rules found by this method are:

```
1.[4<=Fever<=6]                          -> Viral
  [4<=Fever<=6]^[7<=Headache<=14]        -> Bacterial,
2.[2<=Headache<=3]                       -> CT_FIND=normal
  [2<=Headache<=3]^[151<=CSF_PRO<=474] -> CT_FIND=abnormal,
3.[37.6<=BT<=38.8]                       -> EEG_FOCUS=-
  [37.6<=BT<=38.8]^[126<=CSF_PRO<=474] -> EEG_FOCUS=+,
```

The most important characteristics of these rules are that structure of rule pairs is very appealing to medical experts. Especially, the second and third one are very interesting, where CSF_PRO is an important factor for abnormality of CT and EEG findings.

5 Summary

This paper presents comparison of eleven KDD methods by using the common medical dataset on meningoencephaltis. As shown in Section 3 and 4, results induced by each method are different from those by other methods, which suggests that each method should focus on one aspect of knowledge discovery and that combination of all the methods should enhance hypothesis generation in discovery process much more than single rule induction method.

References

1. Adams, R., Victor, M.: Principles of Neurology, 5th edn. McGraw-Hill, New York (1993)
2. Akaike, H.: Information theory and an extention of the maximum likelihood principle. In: Petrov, B., Csaki, F. (eds.) 2nd International Symposium on Information Theory, pp. 267–281. Akadimiai Kiado, Budapes (1973)
3. Durand, M.L., Calderwood, S.B., Weber, D.J., Miller, S.I., Southwick, F.S., Verne, S., Caviness, J., Swartz, M.N.: Acute bacterial meningitis in adults - a review of 493 episodes. New England Journal of Medicine 328, 21–28 (1993)
4. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The kdd process for extracting useful knowledge from volumes of data. CACM 29, 27–34 (1996)
5. Logan, S.A.E., MacMahon, E.: Viral meningitis. British Medical Journal 336, 36 (2008)
6. Rissanen, J.: Stochastic Complexity in Statistical Inquiry. World Scientific, Singapore (1989)
7. Tsumoto, S., et al.: Examination of factors important to predict the prognosis of virus meningoencephalitis. Japanese Kanto Journal of Acute Medicine 12, 710–711 (1991)
8. Shapiro, G., Frawley, W. (eds.): Knowledge Discovery in Databases. AAAI Press, Palo alto (1991)
9. Shavlik, J., Dietterich, T. (eds.): Readings in Machine Learning. Morgan Kaufmann, Palo Alto (1990)
10. Suzuki, E.: Exceptional rule discovery in databases based on information theory. In: Second International Conference on Knowledge Discovery and Data Mining, pp. 275–278. AAAI Press, Menlo Park (1996)
11. Terano, T., Ishino, Y.Y.: Interactive genetic algorithm based feature seelction and its application to marketing data analysis. In: Liu, H., Motada, H. (eds.) Feature Extraction Concstruction and Selection: A Data Mining Perspective, pp. 393–406. Kluwer, Dordrecht (1998)
12. Tsukada, M., Inokuchi, A., Washio, T., Motoda, H.: Comparison of mdlp and aic on discretization of numerical attributes. In: Proceedings of 42nd KBS Meeting: SIG-KBS-9802, pp. 45–52. Japan AI Society (1999) (in Japanese)
13. Tsumoto, S., Ziarko, W.N.S., Tanaka, H.: Knowledge discovery in clinical databases based on variable precision rough set model. In: The Eighteenth Annual Symposium on Computer Applications in Medical Care, pp. 270–274 (1995)
14. Tsumoto, S., Yamaguti, T. (eds.): Proceeding of 42nd KBS meeting: SIG-KBS-9802. Japanese AI Society (1999)
15. Zhong, N., Dong, J., Ohsuga, S.: Data mining based on the generalization distribution table and rough sets. In: Wu, X., Kotagiri, R., Korb, K.B. (eds.) PAKDD 1998. LNCS, vol. 1394, pp. 360–373. Springer, Heidelberg (1998)

Extracting Social Networks Enriched by Using Text

Mathilde Forestier, Julien Velcin, and Djamel Zighed

ERIC Laboratory, University of Lyon 2, France
firstname.name@univ-lyon2.fr

Abstract. Forums on the Internet are an overwhelming source of knowledge considering the number of topics treated and users who participate in these discussions. This volume of data is difficult to comprehend for a person with respect for the large number of posts. Our work proposes a new formal framework for synthesizing information contained in these forums. We extract a social network that reflects reality by extracting multiple relationships between individuals (structural relationship, name and text quotation relationships). These relationships are created from the structure and the content of the discussion. Results show that discovering quotation relationships from forums is not trivial.

1 Introduction

Forums on the Internet connect people who do not know each other and allows them to have a discussion about their subjects of interest. It exists several ways to represent forums. We can represent the discussion using a post graph as in [11], but it seems to be very interesting to also design people interactions. Indeed, we have individuals (recognizable by their pseudonym), who speak to each other through written posts, and who answer mutually using the structure of the forum. But reading discussions shows that authors reply to each others on several ways. Authors can reply by using the website structure. But at the same time, some posts reply to several authors by quoting names while other posts strengthen the structural relationship by quoting text. This analysis of posts made us think of a model with several types of relationships. These relationships reinforce each other. In the light of these observations we defined three types of relationship: the structural, the name and the text quotation relationships.

This paper presents an original approach, which allows to extract a social network with several relationships from the structure and the content of forums. Note that extracting quotations in this kind of text is not trivial due to the poor quality of the writing. In fact, texts usually contain typing errors, misuses of typographical rules, name modifications, to cite a few. Our work has multiple objectives, while our contributions aim to: extract a social network from forums by taking into account several types of relationships from the structure and content of the data and model interactions in English and French languages.

Firstly, we will propose a synthesis of the existing works dedicated to the extraction of social networks. Secondly, we will present the theoretical framework

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 140–145, 2011.

and the system which we have created. Finally, we will describe the method of validation and the results obtained.

2 Related Work

Social network becomes a powerful tool for modeling, understanding and interpreting relationship between individuals. With the emergence of Web 2.0, researchers in new domains (e.g., computer science) use social network to model implicit relationships in the large volumes of data.

Several researchers worked on extracting a social network from a community of researcher [5, 8, 7]. Kautz et al., in 1997 [5] imagined ReferralWeb, a new system to extract relationship from the Web and email archive. Peter Mika [8] implemented Flink, a complete system: from data acquisition (on several sources) to visualization. Note that this system takes into account two types of relationships: the domain of interest shared by two researchers and when two people know each other. Matsuo et al. [7] created a system, called Polyphonet which recognizes four types of possible relationships between two actors. Culotta et al. in [1] created a system that automatically integrates e-mail and Web content to help users to keep up large contact databases to date. Jin et al. [3, 4] think that social network extraction depends on the population that is studied. They studied two different populations: companies and artists. For each of these populations, there are several types of relationships (e.g. alliance relations, litigation for companies). Finally, researchers in the humanities and computer science have been interested in forum representation by social network [2, 6, 12]. To understand the place of individuals in the community, they create a social network using the forum structure (who replies to whom) and used egocentric network: social network focused on an actor and his neighbors at a predefined distance.

Papers dealings with forums only use the structural relationship to model the social network. But relations between individuals are plural [3, 7] and this plurality is not taking account in the existing works. So in this work, we extract several relations to have a better perception of the reality of interactions. Furthermore, using textual content is little used in current works about social network extraction.

3 From Theory to Experiments

People who write in these forums are linked by the structure of the forum, i.e. when an author replies to another one using the forum feature (i.e. 'reply to'). But the structural relationship does not account for implicit relationships contained in the posts. Indeed, some authors reply to several others in the same post, others reply to an author without using the structural relationship, etc. We decide to link authors by three relationships: structural, name and text quotations.

3.1 Theoretical Framework

To model our social network, we first defined three sets:
X: the set of authors $X = \{x_1, ..., x_n\}$ where n is the number of authors.
R: the set of relationships. R is a finite set of three relationships $R = R_{str} \cup R_{text} \cup R_{name}$ respectively structural, text and name quotation relationships.
D: the set of documents (each post represents one document) with $D = \{d_1, ..., d_m\}$ where m is the number of posts in the forum. Note that D is partitioned in T groups where T represents the number of threads. A thread is a part of the forum where the posts reply to each other using the structural relation. Finally, a forum contains a time dimension so $d' < d$ if d' is published before d.

From these three sets, we can define the following mapping:
$authors$: $D \rightarrow X$: a post is written by one actor
$d \mapsto x$
And, knowing that $\delta \in \{str, text\}$ the following binary relationships:
$d_a R_\delta d_b \Leftrightarrow$ document d_a quotes d_b with the δ relationship.
$d_a R_{name} x \Leftrightarrow$ document d_a quotes the pseudonym of x.
And the author relationships:
$x_i R_\delta x_j \Leftrightarrow \exists d_a, d_b \in D \times D / d_a R_\delta d_b$ and $author(d_a) = x_i$ and $author(d_b) = x_j$
$x_i R_{name} x_j \Leftrightarrow \exists d \in D / d R_{name} x_j$ and $author(d) = x_i$

We define a multi-graph G = (X,A) where X represents the set of authors and A the set of directed edges. Each directed edge a_{ijl} represents a reply from an actor x_i to x_j with the relation $r \in \{str, text, name\}$.

3.2 Extracting Quotations

As we saw in the previous subsection, the model has three types of relationships. To automatically extract them, we need to explore the structure and the content of the data.

Extracting Name Quotation. Extracting name quotation is more complex than just looking for the exact name of the authors in the post: the name can be badly spelled in a post or just a part of a compounded name is used. The name can be also written by an abbreviation, a diminutive or a synonym. The pseudo-code in algorithm 1 in Figure 1 shows the approach to extract name quotation. To retrieve the name in spite of the typographical errors we need to compare the post content with previous authors with taking account a margin of error. To this extend, we use the Levenshtein distance [10] (see algorithm 1 at line 8). To increase the results and with the observation that pseudonym is generally nonexistent word, we search each word in the dictionary included in TreeTagger [9] (see algorithm 1 at line 8).

Extracting Text Quotation. Text quotation allows to strengthen the structural relationship. Forum posts make it difficult to retrieve text quotations

because authors do not use or misuse quotations marks. Unlike the name quotation, text quotation is usually used in the same thread. Therefore, to reduce complexity, we seek text quotation in the thread to which the post belongs. The pseudo-code, to extract automatically quotations from posts, is explained in algorithm 2 in Figure 1.

Algorithm 1 Name quotation Extraction

Require: D, X, α, dist, dictionary

for each $d \in D$ **do**
  $X' \Leftarrow \{x \in X, \exists d' \in D / author(d') = x$ and $d' < d\}$
  x $\Leftarrow$ author(d)
  for each $x' \in X'$ **do**
    x' is $(w_1, w_2, ..., w_k)$
    if $\exists w \in x' / \ w \in d$ **then**
      $R \Leftarrow R \cup xR_{name}x'$
    else if $\exists w_i \in x', w_j \in d / dist(w_i, w_j) < \alpha$ and dictionary(w_j) = 'unknown' **then**
      $R \Leftarrow R \cup xR_{name}x'$
    end if
  end for
end for

Algorithm 2 Text quotation Extraction

Require: D, X, ω, R_{str}, sim, ω

for each $d \in D$ **do**
  x $\Leftarrow$ author(d)
  {*Quotation marks*}
  if d contains quotation marks **then**
    s $\Leftarrow$ sentence between quotations marks
    for each d' $\in$ D / d' $<$ d **do**
      x' $\Leftarrow$ author(d')
      if $s \in d'$ **then**
        $R \Leftarrow R \cup xR_{text}x'$
      end if
    end for
  end if
end for

Fig. 1. Algorithms for the name quotation extraction (Algorithm 1) and the text quotation extraction (Algorithm 2)

3.3 System Overview

To extract relationships, the system bases itself on several stages: the first stage parses the HTML page containing the forum. The parser retrieves posts, authors and structural relationship which connects them (which post replies to another). All the information is stored in a database. The second consists of two modules: The extraction of the name quotation relationship (when an author is quoted in the body of a post) and the extraction of the text quotation relationship (when a part of a post is taken back in an other post). The last stage takes all the actors and relationships to create the social network.

4 Results

The lack of labeled data makes validation difficult. For unlabeled data, the use of human evaluator appears as unavoidable. Each forum was evaluated by three different evaluators and they have to write each quotation of name and text he finds out . We keep all the quotations found by at least two evaluators. For each forum, we calculate the recall (number of quotations find by both evaluators and system on number of quotations find by evaluators), the precision (number of quotations find by both evaluators and the system on number of quotations find by the system) and the F-measure (harmonical average of recall and precision to have a performance overview).

The validation concerns only the extraction of quotations. We studied four different forums: two come from a French information website and two from an

Table 1. Recall, precision and F-measure for the four forums

	Text Quotation				Name Quotation			
	French		English		French		English	
Forum	Sarkozy	Roma	Quiet Faith	Diabetes	Sarkozy	Roma	Quiet Faith	Diabetes
recall	0.71	0.85	0.375	0.143	0.86	0.52	0.5	0.67
precision	0.71	0.83	0.5	0.5	0.4	0.65	0.17	1
F-measure	0.71	0.84	0.43	0.222	0.55	0.58	0.25	0.8

American information website[1]. The results shown in Table 1 are discussed in the following paragraphs.

Text quotation. The uneven use of quotation marks raises problem for the automatic recognition of quotations. Certainly, when they are advisedly used, they allow an easy extraction of the quotation. But we can see that the system can not perform as an human. In fact, the worst recall (in table 1) is about 0.143 on Diabetes forum but can be widely better e.g. 0.85 on Roma forum. Just as the recall, precision is better on French forum than on English ones. On English forums, half of the quotations found by our system are also found by the evaluators. Due to the nature of texts studied (the posts of forum contain typings errors, spelling, an approximate follow-up of writing rules), the extraction of quotations is a complex task. Certain users do not put a quotation mark, others open it and do not close it. Other authors group together several passages of a post in the same quotation etc. Finally, the difference of results between the French and English forums come from that French people seem to use more quotation marks.

Name quotation. As for the text quotation, the name extraction does not raise any problems when the name is correctly spelled. But results in Table 1 show that there is no regularity in system performance. In fact, on "Sarkozy" forum, there is a good recall (0.86) which means that the majority of the quotations found by evaluators are also found by the system. But, the recall for this forum is a quite bad (0.4): the system finds a lot of fake quotations. Otherwise, on "diabetes" forum, we have the inverse situation: the precision is about 1 (all the quotations find by the system are correct, but the recall is small (0.67). By analyzing the data, we find typing errors: diminutives (for example call Steam the actor SteamBoater, Pierrr-rôt the actor Pierrrre) and synonymic modifications of the pseudonym (call the actor "the.clam" by "my dear gastropod"). That is how we explain that the system does not reach a score of 100%.

5 Conclusion

This paper presents an original new approach to extract a social network from a Web discussion. The model introduced in this paper, extracts various

[1] www.rue89.com and www.huffingtonpost.com

relationships between the actors using the structure and content of the data. These underlying relationships based on the contents of the discussion allow to enrich the graph (built from the structural relationship) and bring a finer precision of the interactions between people. Results show that it is not easy to extract the quotation relationships. Indeed, the system shows uneven performances between forums due to text quality (posts do not always respect typographic rules, they contain typing errors, etc.). Once the results improve, we will focus on the study of the social network. The definition of the various social roles for the actors is one of the perspectives which seems to us particularly interesting: identify actors and their social roles in a communicating community, appears to us as a main advantage.

References

[1] Culotta, A., McCallum, A., Bekkerman, R.: Extracting social networks and contact information from email and the web (2005)
[2] Fisher, D., Smith, M., Welser, H.: You are who you talk to: Detecting roles in usenet newsgroups. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, HICSS 2006, pp. 59b–59b (2006)
[3] Jin, Y., Matsuo, Y., Ishizuka, M.: Extracting social networks among various entities on the web. The Semantic Web: Research and Applications, 251–266 (2007)
[4] Jin, Y., Matsuo, Y., Ishizuka, M.: Ranking Entities on the Web using Social Network Mining and Ranking Learning. Structure 1, 3 (2008)
[5] Kautz, H., Selman, B., Shah, M.: Referral Web: combining social networks and collaborative filtering. Communications of the ACM 40(3), 63–65 (1997)
[6] Kelly, J., Fisher, D., Smith, M.: Friends, foes, and fringe: norms and structure in political discussion networks. In: Proceedings of the 2006 International Conference on Digital Government Research, pp. 21–24 (May 2006)
[7] Matsuo, Y., Mori, J., Hamasaki, M., Nishimura, T., Takeda, H., Hasida, K., Ishizuka, M.: POLYPHONET: an advanced social network extraction system from the web. Web Semantics: Science, Services and Agents on the World Wide Web 5(4), 262–278 (2007)
[8] Mika, P.: Flink: Semantic web technology for the extraction and analysis of social networks. Web Semantics: Science, Services and Agents on the World Wide Web 3(2-3), 211–223 (2005)
[9] Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: International Conference of New Methods in Language Processing (1994)
[10] Soukoreff, R., MacKenzie, I.: Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic. In: CHI 2001 Extended Abstracts on Human Factors in Computing Systems, pp. 319–320. ACM, New York (2001)
[11] Stavrianou, A.: Modeling and Mining of Web Discussions. Ph.D. thesis, Universit Lumire Lyon 2 (2010)
[12] Welser, H., Gleave, E., Fisher, D., Smith, M.: Visualizing the signatures of social roles in online discussion groups. Journal of Social Structure 8(2) (2007)

Enhancing Navigation in Virtual Worlds through Social Networks Analysis

Hakim Hacid, Karim Hebbar, Abderrahmane Maaradji,
Mohamed Adel Saidi, Myriam Ribière, and Johann Daigremont

Bell Labs France
Centre de Villarceaux Route de Villejust, 91620, Nozay, France
firstname.lastname@alcatel-lucent.com

Abstract. Although Virtual Worlds (VWs) are exponentially gaining popularity, they remain digitalized environments allowing to users only basic interactions and limited experience of life due mainly to the lack of realism and immersion. Thus more and more research initiatives are trying to make VWs more realistic through, for example, the use of haptic equipments and high definition drawing. This paper presents a new contribution towards enhancing VWs realism from the visual perception perspective by performing social networks analysis and conditioning avatars rendering according to social proximities.

1 Introduction

Virtual Worlds (VWs) are computer-simulated environments where the computer presents perceptual stimuli to the user who interacts and manipulates the different elements of that simulated environment. VWs join the increasing important topic related to computer games. This area is attracting more and more researchers, especially in databases, where there are attempts to leverage databases technologies [10] [5] and industry [9] [8] . This increasing interest is also justified by the huge commercial interest of those environments. The most interesting example is certainly "SecondLife"[1] which offers the user locations, objects, etc. which she can manipulate, sell, buy, etc. from other users. Beyond the Web 2.0 dimension, VWs offer opportunities to experience, e.g. tele-presence, to a certain degree, simulate economic or scientific experiments, etc. [1].

Beside making VWs more efficient, inter-operable, easy programmable, there is a big effort in making them more realistic [8] [11] [3]. This goes through making interactions easier and natural with objects and other users in VWs, improving visual perception, reproduction of ambient contexts (e.g., noise, smell, etc.). Basically, all VWs applications share a common drawing and display principle: draw as much details as possible of the elements in the area where the user is located. Thus, the strategy is mainly based on a distance calculation between the current location of the user and the rest of the elements of the concerned area. The drawing operation in VWs is a heavy operation which necessitates many

[1] http://www.secondlife.com/

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 146–152, 2011.

computation resources. This is the main reason, from the perception perspective, VWs are often lacking realism.

In this paper, we consider another emerging problem closely related to the previous one: navigation in the crowd. This problem results from the increasing presence of users in these platforms. Our problem can be formulated as follows: *"how to easily recognize avatars of known persons in a context of a crowded scene?"*. Figure 1 illustrates a crowded VW scene in which several avatars are "living". It is clear that sophisticated image processing techniques need to be used in this context to draw and animate such scene. However, the observation which is directly related to our work is that of the generally limited number of avatars a platform may offer. This implies that many users may use the same avatar to evolve in the VW creating ambiguity and confusion in recognizing avatars. A possible solution is to associate a name to an avatar and show it in the interface, as illustrated in Figure 1. However, on one hand, the user needs to consult the different avatars names to find the right person and probably disturbing the other users, and, on the other hand, this has a negative impact on the natural way of "living" in these worlds which intend to maximize the user immersion and experience by reproducing natural human brain mechanisms.

Fig. 1. Example of a situation in a virtual world

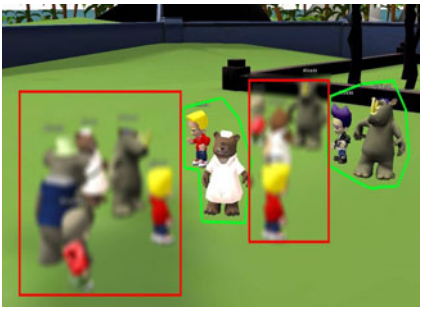


Fig. 2. An example of a standard drawing in a VW: with social-based definition customization

We propose to leverage social networks analysis (SNA) [12] to exploit possible user interactions on the considered world as well as on external on-line social networks as a basis providing a contribution to this problem. The rest of this paper is organized as follows: Section 2 discusses the motivations behind our proposal and the approach we are proposing to enhance virtual worlds realism based on social networks analysis. Section 3 describes a very preliminary observations about the impact of our proposal on the systems performances. Section 4 discusses some related works. We conclude and give some future directions of this work in Section 5.

2 A SNA Approach for Realism Enhancing in VWs

In order for the user to connect to VWs, a client navigator has to be launched on a local machine to display only the VW areas (called *"island"* in SecondLife). A visitor is then on this island where she can meet other avatars representing other users. Let us consider what could happen in the future with the growing success of the VWs: Virtual events such as museum visits and concerts will lead to an increasing number of visitors at the same place and time. A consequence is the difficulty for visitors to drive their avatars in a large crowded scenes and find a specific person. The mechanism of people recognition in a real life crowd [7] needs then to be transposed into VWs. We propose to perform this transposition and make VWs more realistic through the integration of a social dimension into VWs. In fact, users can be part of real communities in the real world, e.g., colleagues, family, Facebook, Twitter, etc. or can create virtual communities and, e.g. meet in some places, plan an (virtual) event, etc. which makes these people linked by a kind of a "virtual" social relationship[2].

To tackle this problem we propose to filter the drawing of details only for specific known persons allowing to distinguish easily user's acquaintances in a crowd of unknown avatars. So, the basic idea is to use an augmented adaptive streaming based technology to discriminate between the persons a user is interested in (their avatars appear to her with all their details/attributes) and the others (who are represented only in basic drawings). An example of what we would obtain is illustrated by Figure 2 representing a scene a user may have when our approach is used. In Figure 2, avatars surrounded with rectangles[3] are drawn with lots of details where avatars surrounded with circles are drawn with low details.

The particularity of this adaptive streaming is that we constrain the drawing with social properties computed between people, meaning that closer an avatar is to me in the VW, better its drawing is. In other words, we inject a social closeness information into the adaptive streaming for adapting the drawing. It should be noted that this translates somehow the mechanism of the human cognition as discussed in the following. In fact, when a person (v_1) is visiting a place for example, this person doesn't care about persons who are near to her (in terms of physical distance) and thus doesn't focus on them. v_1 is implicitly ignoring the surrounding persons. In the other case, when v_1 recognizes another person, say v_2, even in a crowd, v_1 focuses on v_2 and the rest of the crowd becomes meaningless for her.

If we translate the example to the proposed solution, in the first case, the social proximity between v_1 and the crowd is not very high and thus it is not useful to consume resources to make clearer drawings of the crowd. However, in the second case, the most important consideration to the user is to let her

[2] We focus here on digital interactions, i.e., interactions that occur on, e.g., social networking sites or VWs.

[3] Note that rectangles and circles are drawn for illustration only in this paper and not on the system.

identify known persons quickly and easily. Since v_2 is socially close to v_1, more efforts should be made to draw in a clearer way the corresponding avatars. To tackle this, we make use of a simple but powerful data model which captures social interactions independently of the social networks.

We define a social network is a directed, weighted, and labeled graph capturing interactions between people. These interactions may happen in on-line platforms or, in a general way, user's life. Nodes of this graph represent people with their properties and the links the interactions between those people. Let G denotes a social graph of, say persons, defined as $G(V, A, W(A))$ with $V = \{v_1, ..., v_n\}$ representing a set of n nodes of the graph (corresponding to a set of persons in this case). A represents a set of arcs linking nodes of the graph. It should be noted that each node could be associated to different real accounts. This detail is omitted for clarity matters. G is a directed graph, the following property apply then for each arc: $\forall v_i, v_j \in V^2, (v_i, v_j) \neq (v_j, v_i)$ Since G is a directed and weighted graph, we can define a function $\omega : V \times V \rightarrow R^+$ such that: $\forall v_i, v_j \in V^2, if\ (v_i, v_j) \in A\ then\ \omega(v_i, v_j) \in R^+$. Thus, ω associates a weight for all the couples of nodes which have a common interactions.

A social proximity can be defined and calculated with several ways and depending on the context of the analysis and the level of details the situation requires. Many variants using different parameters have been used [12]. In our case, we consider the social proximity either (i) explicitly declared by the user or (ii) calculated on shared activities (e.g., discussions, media exchanges, etc.) on different social networking sites. Let's consider m social networking sites $\{s_1, s_2, ..., s_m\}$ and the user $v_i \in V$ is connected to $k \leq m$ in which the user discusses with her friends, exchanged photos, recommends content and products, etc. The user has an activity indicator function $A_l : U \times U \rightarrow R^+, (l = 1, ..., k)$ corresponding to each social networking site. Let's now consider a virtual world VW in which the user shares virtual places, meets with other friends around specific events, etc. To calculate the social proximity, we take advantage of the activities on the different social networking sites. Formula 1 illustrates the way the social proximity is calculated where $ACT(s_t, v_i)$ is the overall activity of user v_i in site s_t which is intended to measure, e.g., the frequency of use and the importance of a given social networking website to a user.

$$d(v_i, v_j) = [\sum_{l=1}^{k} (A_l(v_i, v_j)/ACT(s_t, v_i))]/k \quad (1)$$

To be able to translate the calculated social proximity into a display constraint, we propose to introduce the notion of *social display definition* (SDD). A SDD is a set of three basic zones that constraint the display definition of an avatar according to a quantitative social proximity (i.e., Formula 1). The zones are defined according to a social proximity, or social layers as considered in social networks. The more there is a move toward the center, greater is the social proximity and clearer is the drawing.

Finally, our proposal aims at showing the details of persons (i.e. avatars) who are socially close to us in VWs while hiding these details for unknown persons. From the system perspective, this could be considered as an extension in VWs streaming systems. By considering this new approach, we define two complementary modes: (i) *Normal mode*: constitutes the default one used in the current systems, and (ii) *Advanced mode*: considers the social proximity. It is used to help people easily navigate in their virtual world. Moreover, it is useful in exploration of VWs to organize relatives. These two modes can cohabit in the same system and (i) offer the user the ability to better exploit her social proximity with relatives in her social networks and (ii) enables optimizations on the system since the drawing process is conditioned. In the following section, we describe the formal evaluation performed on the proposed approach to understand it's potential impacts on the system.

Table 1. Social display definition (SDD)

Avatar Type	*Zone*	*Display definition*	*Proximity Threshold*
Socially close avatar	Z1	Very High Definition(VHD)	≥ 0.75
	Z2	High Definition (HD)	≥ 0.4 and < 0.75
	Z3	Medium Definition (MD)	< 040
Physically close avatars	Z1	Enhanced Medium Definition (EMD))	$< 5m$
	Z2	Medium Definition (MD)	$\geq 5m$ and $\leq 10m$
	Z3	Low Definition (LD)	$> 10m$

3 Evaluation and Preliminary Results

Our proposal enables the user to: (i) explicitly recognize socially close avatars in a crowded location and (ii) implicitly, leverage all user's activities on the different social networks. The proposal has been implemented as an extension of the Solipsis platform [6]. This is interesting for the end-user but still needs to be evaluated and confronted to the end-user for further improvements. In this section, we present some preliminary results regarding the impact of our approach on system's resources. Our assumption in the beginning of this work is that our method could optimize computation resources on the different peers since we are supposed to manage less detailed avatars. We have performed another more technical experiment regarding the behavior of the proposed approach. The idea here was mainly to measure the potential impact of our proposal on the performances of the system. To perform this experiment we have followed a particular protocol described hereafter: We have used *Solipsis*, as peer to peer virtual world and built a network of 4 machines (*HP Elite Book 8730, Intel core Duo 2.8 Ghz and 3 Go of memory*). On each machine we launch 5 avatars (20 avatars for the whole network). The server (i.e., worlds coordinator) is launched only on one machine and the rest of the machines have only clients. Each avatar is associated with a set of 4 to 5 social relatives in the data set through an explicit declaration of the link.

We first capture the processor and the memory activity until we launch all the avatars. Within a regular interval, we remove an avatar from the world (i.e., $d(v_i, v_j) = +\infty$) and we observe the behavior of the processor and the memory use of the process. We repeat this task until all the avatars have been removed. Our observation is that, although our proposal doesn't improve the resources management, it doesn't worsen it. The reason is that we greatly depend on the internal details of the existing implementation. The most important evaluation that we need to perform is certainly an end-user evaluation. These preliminary system experiments will serve as a basis for a better implementation in the future for the integration of the proposed method in the next release of *Solipsis*.

4 Related Work

Our work is related to virtual worlds which, as discussed before, are becoming more and more interesting in the digital life of users. VWs are considered of interest from both industrial and research perspective. From the industrial perspective, most of the efforts are performed towards the integration of a 3D visualization mechanism into, e.g., devices. This includes 3D TV [9], 3D API standardization and interoperability mechanisms between devices and virtual environments [8]. From this perspective, our contribution is intended to enhance avatars drawing in order to help replicating human brain recognition mechanism.

From the research perspective, VWs have been addressed in different communities. The most related ones are certainly: (i) imagery and (ii) databases. From the imagery perspective, the objective is to find new techniques for enhancing the rendering of virtual environments while optimizing the calculation resources [2] [4]. Our work focuses on display adaptation rather than display improvement, so we are not dealing directly with imagery algorithms. Finally, from the database community perspective, we may consider two visions: (1) proposing new scripting languages for making it easy to implement processes inside virtual worlds [10] [5] or (2) increasing the interoperability and the openness of virtual worlds by bringing external data sources inside these virtual worlds [3]. Our proposal is related to databases since we provide information from outside VWs and thus increase the openness of these environments. Our work is strongly dealing with data integration since we bring data from heterogeneous sources, aggregate them and inject them into the VWs for rendering adaptation making it closely related to [3].

5 Conclusion and Future Work

We discussed in this paper a new approach for enhancing realism in VWs. We have proposed a new approach based on the exploitation of social networks analysis for controlling avatars drawings depending on the social proximity the real user (i.e., who is using the avatar) has with other avatars (i.e., users) in other

social networks. The benefits of the proposed approach are: (i) translation of the human brain recognition mechanism into VWs thanks to social proximities, (ii) the optimization of resources usage thanks to drawing control, and (iii) making VWs open to external sources instead of being closed systems. As a future work, many directions could be considered but we plan to focus on the evaluation of our proposal directly with users since we are targeting end-users.

Acknowledgment

This work is performed in the Metaverse project and is partly supported by the DGCIS directorate from the French Ministry of Economy, Industry and Employment as innovation aid under grant number: ITEA 2 - 07016.

References

1. Biocca, F., Levy, M.R. (eds.): Communication in the age of virtual reality. L. Erlbaum Associates Inc., Hillsdale (1995)
2. Boukerche, A., Feng, J., de Araujo, R.B.: 3d image-based rendering technique for mobile handheld devices
3. Campi, A., Gottlob, G., Hoye, B.: Wormholes of communication: Interfacing virtual worlds and the real world. In: AINA 2009, pp. 2–9. IEEE Computer Society, Washington, DC, USA (2009)
4. Chang, C.-F., Ger, S.-H.: Enhancing 3D graphics on mobile devices by image-based rendering. In: Chen, Y.-C., Chang, L.-W., Hsu, C.-T. (eds.) PCM 2002. LNCS, vol. 2532, pp. 1105–1111. Springer, Heidelberg (2002)
5. Demers, A.J., Gehrke, J., Koch, C., Sowell, B., White, W.M.: Database research in computer games. In: SIGMOD Conference, pp. 1011–1014 (2009)
6. Frey, D., Royan, J., Piegay, R., Kermarrec, A., Anceaume, E., Le Fessant, F.: Solipsis: A decentralized architecture for virtual environments. In: MMVE 2008 (March 2008)
7. Inui, T.: Mechanisms of action generation and recognition in the human brain. In: ICKS 2007, pp. 45–52. IEEE Computer Society, Washington, DC, USA (2007)
8. MPEG-V. Information exchange with virtual worlds, http://mpeg.chiariglione.org/working_documents.htm#mpeg-v (visit March 2011)
9. Onural, L., Sikora, T., Osterman, J., Smolic, A., Cyvanlar, M.R., Watson, J.: An assement of 3dtv technologies. In: Proc. NAB 2006, pp. 456–467 (2006)
10. Sowell, B., Demers, A.J., Gehrke, J., Gupta, N., Li, H., White, W.M.: From declarative languages to declarative processing in computer games. In: CIDR (2009)
11. Timmerer, C., Gelissen, J., Waltl, M., Hellwagner, H.: Interfacing with virtual worlds. In: Proceedings of the NEM Summit 2009, Saint-Malo, France, September 28-30 (2009)
12. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences), 1st edn. Cambridge University Press, Cambridge (1994)

Learning Diffusion Probability Based on Node Attributes in Social Networks

Kazumi Saito[1], Kouzou Ohara[2], Yuki Yamagishi[1], Masahiro Kimura[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[3] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. Information diffusion over a social network is analyzed by modeling the successive interactions of neighboring nodes as probabilistic processes of state changes. We address the problem of estimating parameters (diffusion probability and time-delay parameter) of the probabilistic model as a function of the node attributes from the observed diffusion data by formulating it as the maximum likelihood problem. We show that the parameters are obtained by an iterative updating algorithm which is efficient and is guaranteed to converge. We tested the performance of the learning algorithm on three real world networks assuming the attribute dependency, and confirmed that the dependency can be correctly learned. We further show that the influence degree of each node based on the link-dependent diffusion probabilities is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the true link-dependent diffusion probabilities.

1 Introduction

The growth of Internet has enabled to form various kinds of large-scale social networks, through which a variety of information, e.g. news, ideas, hot topics, malicious rumors, etc. spreads in the form of "word-of-mouth" communications, and it is noticeable to observe how much they affect our daily life style. The spread of information has been studied by many researchers [15,14,4,1,12,7,9]. The information diffusion models widely used are the *independent cascade (IC)* [2,5,7] and the *linear threshold (LT)* [21,22] models. They have been used to solve such problems as the *influence maximization problem* [5,8] and the *contamination minimization problem* [7,20]. These two models focus on different information diffusion aspects. The IC model is sender-centered (push type) and each active node *independently* influences its inactive

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 153–162, 2011.

neighbors with given diffusion probabilities. The LT model is receiver-centered (pull type) and a node is influenced by its active neighbors if their total weight exceeds the threshold for the node.

What is important to note is that both models have parameters that need be specified in advance: diffusion probabilities for the IC model, and weights for the LT model. However, their true values are not known in practice. This poses yet another problem of estimating them from a set of information diffusion results that are observed as time-sequences of influenced (activated) nodes. This falls in a well defined parameter estimation problem in machine learning framework. Given a generative model with some parameters and the observed data, it is possible to calculate the likelihood that the data are generated and the parameters can be estimated by maximizing the likelihood. To the best of our knowledge, we are the first to follow this line of research. We addressed this problem for the IC model [16] and devised the iterative parameter updating algorithm.

The problem with both the IC and LT models is that they treat the information propagation as a series of state changes of nodes and the changes are made in a synchronous way, which is equivalent to assuming a discrete time step. However, the actual propagation takes place in an asynchronous way along the continuous time axis, and the time stamps of the observed data are not equally spaced. Thus, there is a need to extend both models to make the state changes asynchronous. We have, thus, extended both the models to be able to simulate asynchronous time delay (the extended models are called AsIC and AsLT models) and showed that the same maximum likelihood approach works nicely [17,18,19] and recently extended the same approach to opinion propagation problem using the value-weighted voter model with multiple opinions [10]. There are other works which are close to ours that also attempted to solve the similar problem by maximizing the likelihood [3,13], where the focus was on inferring the underlying network. In particular, [13] showed that the problem can effectively be transformed to a convex programming for which a global solution is guaranteed.

In this paper we also address the same problem using the AsIC model, but what is different from all of the above studies is that we try to learn the dependency of the diffusion probability and the time-delay parameter on the node attributes rather than learn it directly from the observed data. In reality the diffusion probability and the time-delay parameter of a link in the network must at least be a function of the attributes of the two connecting nodes, and ignoring this property does not reflect the reality. Another big advantage of explicitly using this relationship is that we can avoid overfitting problem. Since the number of links is much larger than the number of nodes even if the social network is known to be sparse, the number of parameters to learn is huge and we need prohibitively large amount of data to learn each individual diffusion probability separately. Because of this difficulty, many of the studies assumed that the parameter is uniform across different links or it depends only on the topic (not on the link that the topic passes through). Learning a function is much more realistic and does not require such a huge amount of data.

We show that the parameter updating algorithm is very efficient and is guaranteed to converge. We tested the performance of the algorithm on three real world networks assuming the attribute dependency of the parameters. The algorithm can correctly estimate both the diffusion probability and the time-delay parameter by way of node

attributes through a learned function, and we can resolve the deficiency of uniform parameter value assumption. We further show that the influence degree of each node based on the link-dependent diffusion probabilities (via learned function) is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the link-dependent diffusion probabilities, indicating that the uniform diffusion probability assumption is not justified if the true diffusion probability is link-dependent.

2 Diffusion Model

2.1 AsIC Model

To mathematically model the information diffusion in a social network, we first recall the AsIC model according to [19], and then extend it to be able to handle node attributes. Let $G = (V, E)$ be a directed network without self-links, where V and E ($\subset V \times V$) stand for the sets of all the nodes and links, respectively. For each node $v \in V$, let $F(v)$ be the set of all the nodes that have links from v, *i.e.*, $F(v) = \{u \in V;\ (v, u) \in E\}$, and $B(v)$ be the set of all the nodes that have links to v, *i.e.*, $B(v) = \{u \in V;\ (u, v) \in E\}$. We say a node is *active* if it has been influenced with the information; otherwise it is inactive. We assume that a node can switch its state only from inactive to active.

The AsIC model has two types of parameter $p_{u,v}$ and $r_{u,v}$ with $0 < p_{u,v} < 1$ and $r_{u,v} > 0$ for each link $(u, v) \in E$, where $p_{u,v}$ and $r_{u,v}$ are referred to as the diffusion probability and the time-delay parameter through link (u, v), respectively. Then, the information diffusion process unfolds in continuous-time t, and proceeds from a given initial active node in the following way. When a node u becomes active at time t, it is given a single chance to activate each currently inactive node $v \in F(u)$: u attempts to activate v if v has not been activated before time $t + \delta$, and succeeds with probability $p_{u,v}$, where δ is a delay-time chosen from the exponential distribution[1] with parameter $r_{u,v}$. The node v will become active at time $t + \delta$ if u succeed. The information diffusion process terminates if no more activations are possible.

2.2 Extension of AsIC Model for Using Node Attributes

In this paper, we extend the AsIC model to explicitly treat the attribute dependency of diffusion parameter through each link. Each node can have multiple attributes, each of which is either nominal or numerical. Let v_j be a value that node v takes for the j-th attribute, and J the total number of the attributes. For each link $(u, v) \in E$, we can consider the J-dimensional vector $\boldsymbol{x}_{u,v}$, each element of which is calculated by some function of u_j and v_j, *i.e.*, $x_{u,v,j} = f_j(u_j, v_j)$. Hereafter, for the sake of convenience, we consider the augmented $(J + 1)$-dimensional vector $\boldsymbol{x}_{u,v}$ by setting $x_{u,v,0} = 1$ as the link attributes. Then we propose to model both the diffusion probability $p_{u,v}$ and the time-delay parameter $r_{u,v}$ for each link $(u, v) \in E$ by the following formulae[2]:

$$p_{u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x}_{u,v})}, \quad r_{u,v} = r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) = \exp(\boldsymbol{\phi}^T \boldsymbol{x}_{u,v}), \qquad (1)$$

[1] We chose a delay-time from the exponential distribution in this paper for the sake of convenience, but other distributions such as power-law and Weibull can be employed.

[2] Note that both are simple and smooth functions of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ that guarantee $0 < p < 1$ and $r > 0$.

where $\boldsymbol{\theta}^T = (\theta_0, \cdots, \theta_J)$ and $\boldsymbol{\phi}^T = (\phi_0, \cdots, \phi_J)$ are the $(J+1)$-dimensional parameter vectors for diffusion probability and time-delay parameter, respectively. Note here that θ_0 and ϕ_0 correspond to the constant terms, and $\boldsymbol{\theta}^T$ stands for a transposed vector of $\boldsymbol{\theta}$.

Although our modeling framework does not depend on a specific form of function f_j, we limit the form to be the following: $x_{u,v,j} = \exp(-|u_j - v_j|)$ if the j-th node attribute is numerical; $x_{u,v,j} = \delta(u_j, v_j)$ if the j-th node attribute is nominal, where $\delta(u_j, v_j)$ is a delta function defined by $\delta(u_j, v_j) = 1$ if $u_j = v_j$; $\delta(u_j, v_j) = 0$ otherwise. Intuitively, the more similar u_j and v_j are, that is, the closer their attribute values are to each other, the larger the diffusion probability $p_{u,v}$ is if the corresponding parameter value θ_j is positive, and the smaller if it is negative. We can see the similar observation for the time-delay parameter $r_{u,v}$.

3 Learning Problem and Method

We consider an observed data set of M independent information diffusion results, $\mathcal{D}_M = \{D_m;\ m = 1, \cdots, M\}$. Here, each D_m represents a sequence of obserbation. It is given by a set of pairs of active node and its activation time, $D_m = \{(u, t_{m,u}),\ (v, t_{m,v}),\ \cdots\}$, and called the mth diffusion result. These sequences may partially overlap, *i.e.*, a node may appear in more than one sequence, but are treated separately according to the AsIC model. We denote by $t_{m,v}$ the activation time of node v for the mth diffusion result. Let T_m be the observed final time for the mth diffusion result. Then, for any $t \leq T_m$, we set $C_m(t) = \{v \in V;\ (v, t_{m,v}) \in D_m,\ t_{m,v} < t\}$. Namely, $C_m(t)$ is the set of active nodes before time t in the mth diffusion result. For convenience sake, we use C_m as referring to the set of all the active nodes in the mth diffusion result. For each node $v \in C_m$, we define the following subset of parent nodes, each of which had a chance to activate v, i.e., $\mathcal{B}_{m,v} = B(v) \cap C_m(t_{m,v})$.

3.1 Learning Problem

According to Saito et al. [17], we define the probability density $\mathcal{X}_{m,u,v}$ that a node $u \in \mathcal{B}_{m,v}$ activates the node v at time $t_{m,v}$, and the probability $\mathcal{Y}_{m,u,v}$ that the node v is not activated by a node $u \in \mathcal{B}_{m,v}$ within the time-period $[t_{m,u}, t_{m,v}]$.

$$\mathcal{X}_{m,u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) \exp(-r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})). \tag{2}$$

$$\mathcal{Y}_{m,u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) \exp(-r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})) + (1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})). \tag{3}$$

Then, we can consider the following probability density $h_{m,v}$ that the node v is activated at time $t_{m,v}$:

$$h_{m,v} = \sum_{u \in \mathcal{B}_{m,v}} \mathcal{X}_{m,u,v} \left(\prod_{z \in \mathcal{B}_{m,v} \setminus \{u\}} \mathcal{Y}_{m,z,v} \right) = \prod_{z \in \mathcal{B}_{m,v}} \mathcal{Y}_{m,z,v} \sum_{u \in \mathcal{B}_{m,v}} \mathcal{X}_{m,u,v} (\mathcal{Y}_{m,u,v})^{-1}. \tag{4}$$

Next, we consider the following probability $g_{m,v,w}$ that the node w is not activated by the node v before the observed final time T_m.

$$g_{m,v,w} = p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta}) \exp(-r(\boldsymbol{x}_{v,w}, \boldsymbol{\phi})(T_m - t_{m,v})) + (1 - p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta})). \tag{5}$$

Here we can naturally assume that each information diffusion process finished sufficiently earlier than the observed final time, i.e., $T_m \gg \max\{t_{m,v}; (v, t_{m,v}) \in D_m\}$. Thus, as $T_m \to \infty$ in Equation (5), we can assume

$$g_{m,v,w} = 1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}). \tag{6}$$

By using Equations (4) and (6), and the independence properties, we can define the likelihood function $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ by

$$\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi}) = \log \prod_{m=1}^{M} \prod_{v \in C_m} \left(h_{m,v} \prod_{w \in F(v) \setminus C_m} g_{m,v,w} \right). \tag{7}$$

In this paper, we focus on Equation (6) for simplicity, but we can easily modify our method to cope with the general one (i.e., Equation (5)). Thus, our problem is to obtain the values of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, which maximize Equation (7). For this estimation problem, we derive a method based on an iterative algorithm in order to stably obtain its solution.

3.2 Learning Method

Again, according to Saito et al. [17], we introduce the following variables to derive an EM like iterative algorithm.

$$\mu_{m,u,v} = \mathcal{X}_{m,u,v}(\mathcal{Y}_{m,u,v})^{-1} \Big/ \sum_{z \in \mathcal{B}_{m,v}} \mathcal{X}_{m,z,v}(\mathcal{Y}_{m,z,v})^{-1}.$$
$$\eta_{m,u,v} = p_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})) \,/\, \mathcal{Y}_{m,u,v}.$$
$$\xi_{m,u,v} = \mu_{m,u,v} + (1 - \mu_{m,u,v})\eta_{m,u,v}.$$

Let $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\phi}}$ be the current estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. Similarly, let $\bar{\mathcal{X}}_{m,u,v}$, $\bar{\mathcal{Y}}_{m,u,v}$, $\bar{\mu}_{m,u,v}$, $\bar{\eta}_{m,u,v}$, and $\bar{\xi}_{m,u,v}$ denote the values of $\mathcal{X}_{m,u,v}$, $\mathcal{Y}_{m,u,v}$, $\mu_{m,u,v}$, $\eta_{m,u,v}$, and $\xi_{m,u,v}$ calculated by using $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\phi}}$, respectively.

From Equations (4), (6) and (7), we can transform our objective function $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ as follows:

$$\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi}) = Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) - \mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}), \tag{8}$$

where $Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is defined by

$$Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) + Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$$
$$Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = \sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} (\bar{\xi}_{m,u,v} \log p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) + (1 - \bar{\xi}_{m,u,v}) \log(1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})) \right.$$
$$\left. + \sum_{w \in F(v) \setminus C_m} \log(1 - p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta})) \right), \tag{9}$$
$$Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} (\bar{\mu}_{m,u,v} \log r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) - \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})), \tag{10}$$

and $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is defined by

$$\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} (\bar{\mu}_{m,u,v} \log \mu_{m,u,v}$$
$$+ (1 - \bar{\mu}_{m,u,v})(\bar{\eta}_{m,u,v} \log \eta_{m,u,v} + (1 - \bar{\eta}_{m,u,v}) \log(1 - \eta_{m,u,v})). \tag{11}$$

Since $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is maximized at $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$ and $\boldsymbol{\phi} = \bar{\boldsymbol{\phi}}$ from Equation (11), we can increase the value of $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ by maximizing $Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ (see Equation (8)).

We can maximize Q by independently maximizing Q_1 and Q_2 with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. Here, by noting the definition of $p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})$ described in Equation (1), we can derive the gradient vector and the Hessian matrix of Q_1 as follows:

$$\frac{\partial Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\theta}} = \sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} (\bar{\xi}_{m,u,v} - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})) \boldsymbol{x}_{u,v} - \sum_{w \in F(v) \setminus C_m} p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta}) \boldsymbol{x}_{v,w} \right), \tag{12}$$

$$\frac{\partial^2 Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} = - \sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} \zeta_{u,v} \boldsymbol{x}_{u,v} \boldsymbol{x}_{u,v}^T + \sum_{w \in F(v) \setminus C_m} \zeta_{v,w} \boldsymbol{x}_{v,w} \boldsymbol{x}_{v,w}^T \right), \tag{13}$$

where $\zeta_{u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})(1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}))$. We see that the Hessian matrix of Q_1 is non-positive definite, and thus, we can obtain the optimal solution of Q_1 by using the Newton method. Similarly, we can derive the gradient vector and the Hessian matrix of Q_2 as follows:

$$\frac{\partial Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\phi}} = \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} (\bar{\mu}_{m,u,v} - \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})) \boldsymbol{x}_{u,v}, \tag{14}$$

$$\frac{\partial^2 Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\phi} \partial \boldsymbol{\phi}^T} = - \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u}) \boldsymbol{x}_{u,v} \boldsymbol{x}_{u,v}^T. \tag{15}$$

The Hessian matrix of Q_2 is also non-positive definite, and we can obtain the optimal solution by Q_2. Note that we can regard our estimation method as a variant of the EM algorithm. We want to emphasize here that each time iteration proceeds the value of the likelihood function never decreases and the iterative algorithm is guaranteed to converge due to the convexity of Q.

4 Experimental Evaluation

We experimentally evaluated our learning algorithm by using synthetic information diffusion results generated from three large real world networks. Due to the page limitation, here we show only the results for the parameter vector $\boldsymbol{\theta}$, but we observed the similar results for the parameter vector $\boldsymbol{\phi}$. Note that $\boldsymbol{\phi}$ does not affect the influence degree used in our evaluation described later.

4.1 Dataset

We adopted three datasets of large real networks, which are all bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [7], and has 12,047 nodes and 79,920 directed links (the blog network). The second one is a network derived from the Enron Email Dataset [11] by extracting the senders and the recipients and linking those that had bidirectional communications and there were 4,254 nodes and 44,314 directed links (the Enron network). The last one is a network

Table 1. Absolute errors of estimated parameter values for each network. Values in parentheses are the assumed true values.

network	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}
Blog	0.0380	0.0587	0.1121	0.0941	0.0874	0.0873	0.0419	0.0723	0.0398	0.0400	0.0378
	(-2.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)
Enron	0.0371	0.0465	0.1152	0.0637	0.0758	0.0692	0.0382	0.0831	0.0400	0.0370	0.0385
	(-3.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)
Wikipedia	0.0485	0.0455	0.1505	0.0945	0.0710	0.0897	0.0444	0.1079	0.0438	0.0458	0.0434
	(-4.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)

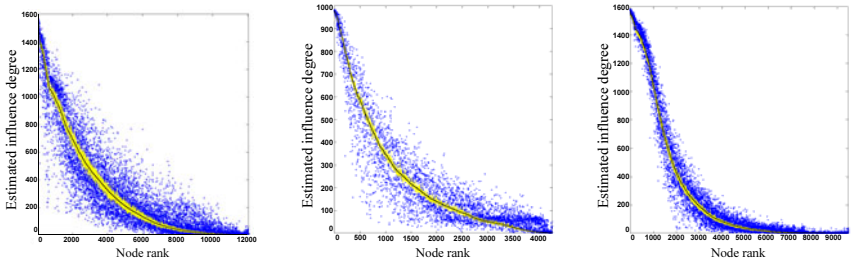
of people that was derived from the "list of people" within Japanese Wikipedia, used in [6], which has 9,481 nodes and 245,044 directed links (the Wikipedia network).

For each network, we generated synthetic information diffusion results in the following way: 1) artificially generate node attributes and determine their values in a random manner; 2) determine a parameter vector $\boldsymbol{\theta}$ which is assumed to be true; and then 3) generate 5 distinct information diffusion results, $\mathcal{D}_5 = \{D_1, \cdots, D_5\}$, each of which starts from a randomly selected initial active node, and contains at least 10 active nodes by the AsIC model mentioned in section 2.2. We generated a total of 10 attributes for every node in each network: 5 ordered attributes, each with a non-negative integer less than 20, and 5 nominal attributes, each with either 0, 1, or 2. The true parameter vector $\boldsymbol{\theta}$ was determined so that, according to [5], the average diffusion probability derived from the generated attribute values and $\boldsymbol{\theta}$ becomes smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. We refer to thus determined values as base values. The resulting average diffusion probability was 0.142 for the blog network, 0.062 for the Enron network, and 0.026 for the Wikipedia network, respectively.

4.2 Results

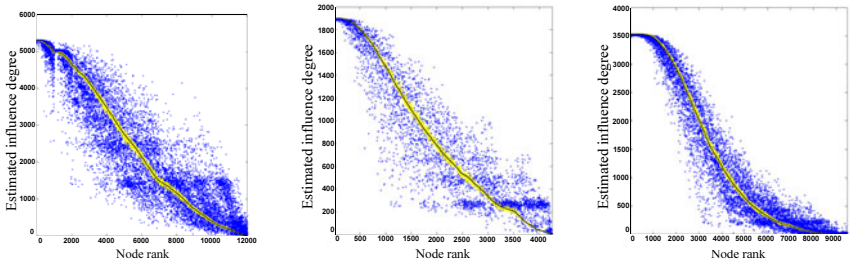
First, we examined the accuracy of parameter values $\hat{\boldsymbol{\theta}}$ estimated by our learning algorithm. Table 1 shows the absolute error $|\theta_i - \hat{\theta}_i|$ for each network which is the average over 100 trials, each obtained from a different $\mathcal{D}_5$ (we generated $\mathcal{D}_5$ 100 times.) where the values in the parentheses are true parameter values. On average, the absolute error of each parameter is 0.0645, 0.0586, and 0.0714 for the blog, Enron, and Wikipedia network, and their standard deviations are 0.0260, 0.0243, and 0.0338, respectively. This result shows that our learning method can estimate parameter values with very high accuracy regardless of networks. Note that $\theta_3, \theta_4, \theta_5, \theta_8, \theta_9$, and θ_{10} are set to 0. This is different from limiting the number of attributes to 4. The average computation time that our learning algorithm spent to estimate the parameter values was 2.96, 6.01, and 28.24 seconds for the blog, Enron, and Wikipedia network, respectively, which means that our learning method is very efficient (machine used is Intel(R) Xeon(R) CPU W5590 @3.33GHz with 32GB memory). Note that, from the derivation in Section 3.2, the computation time depends on the density of the network, i.e. the number of parents of a node.

Next, we evaluated our learning algorithm in terms of the influence degree of each node v which is defined as the expected number of active nodes after the information diffusion is over when v is chosen to be the initial active node. In this experiment,



(a) Blog (b) Enron (c) Wikipedia

Fig. 1. Comparison of three influence degrees σ (black solid line), $\hat{\sigma}$ (yellow marker) and $\bar{\sigma}$ (blue marker) for one particular run, randomly selected from the 100 independent trials in case that the diffusion probabilities are the base values



(a) Blog (b) Enron (c) Wikipedia

Fig. 2. Comparison of three influence degrees σ (black solid line), $\hat{\sigma}$ (yellow marker) and $\bar{\sigma}$ (blue marker) for one particular run, randomly selected from the 100 independent trials in case that the diffusion probabilities are larger than the base values

we derived the influence degree of each node by computing the empirical mean of the number of active nodes obtained from 1,000 independent runs which are based on the bond percolation technique described in [9]. Here, we compared the influence degree $\hat{\sigma}(v)$ of a node v which was derived using the parameter values estimated by our learning algorithm with the influence degree $\bar{\sigma}(v)$ which was derived by a naive way that uses the uniform diffusion probability approximated by averaging the true link-dependent diffusion probabilities.

Figure 1 presents three influence degrees σ, $\hat{\sigma}$, and $\bar{\sigma}$ for each node v for one particular run, randomly chosen from the 100 independent trials, where σ denotes the influence degree derived using the true link-dependent diffusion probability. The nodes are ordered according to the estimated true rank of influential degree. From these figures, we can observe that the difference between σ (solid line) and $\hat{\sigma}$ (yellow) is quite small, while the difference between σ and $\bar{\sigma}$ (blue) is very large and widely fluctuating. In fact, for $\hat{\sigma}$, the average of the absolute error defined as $|\hat{\sigma}(v) - \sigma(v)|$ over all nodes and all trials is 13.91, 6.80, and 8.32 for the blog, Enron, and Wikipedia network, and their standard deviations are 16.41, 7.08, and 11.31, respectively. Whereas, for $\bar{\sigma}$, the corresponding average of $|\bar{\sigma}(v) - \sigma(v)|$ is 77.75, 54.51, and 35.02, and their standard

deviations are 96.12, 57.80, and 51.84, respectively. Even in the best case for $\bar{\sigma}$ (the Wikipedia network), the average error for $\bar{\sigma}$ is about 4 times larger than that for $\hat{\sigma}$.

We further investigated how the error changes with the diffusion probabilities. Figure 2 is the results where the diffusion probabilities are increased, *i.e.*, larger influence degrees expected. To realize this, θ_0 is increased by 1 for each network, i.e. $\theta_0 = -1, -2$, and -3 for the blog, Enron, and Wikipedia network, respectively, which resulted in the corresponding average diffusion probability of 0.28, 0.14, and 0.063, respectively. It is clear that the difference between σ and $\hat{\sigma}$ remains very small, but the difference between σ and $\bar{\sigma}$ becomes larger than before (Fig. 1). Actually, for $\hat{\sigma}$, the average (standard deviation) of the absolute error over all nodes and all trials is 47.95 (28.03), 13.27 (12.30), and 15.11 (16.25) for the blog, Enron, and Wikipedia network, respectively, while, for $\bar{\sigma}$, the corresponding average (standard deviation) is 518.94 (502.05), 162.56 (159.40), and 163.51 (205.17), respectively. These results confirm that $\hat{\sigma}$ remains close to the true influence degree regardless of the diffusion probability p, while $\bar{\sigma}$ is very sensitive to p.

Overall, we can say that our learning algorithm is useful for estimating the influence degrees of nodes in a network, provided that we have some knowledge of dependency of diffusion probability on the selected attributes. It can accurately estimate them from a small amount of information diffusion results and avoid the overfitting problem.

5 Conclusion

Information diffusion over a social network is analyzed by modeling the cascade of interactions of neighboring nodes as probabilistic processes of state changes. The number of the parameters in the model is in general as many as the number of nodes and links, and amounts to several tens of thousands for a network of node size about ten thousands. In this paper, we addressed the problem of estimating link-dependent parameters of probabilistic information diffusion model from a small amount of observed diffusion data. The key idea is not to estimate them directly from the data as has been done in the past studies, but to learn the functional dependency of the parameters on the small number of node attributes. The task is formulated as the maximum likelihood estimation problem, and an efficient parameter update algorithm that guarantees the convergence is derived. We tested the performance of the learning algorithm on three real world networks assuming a particular class of attribute dependency, and confirmed that the dependency can be correctly learned even if the number of parameters (information diffusion probability of each link in this paper) is several tens of thousands. We further showed that the influence degree of each node based on the link-dependent diffusion probabilities is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the true link-dependent diffusion probabilities. This indicates that use of uniform diffusion probability is not justified if the true distribution is non-uniform, and affects the influential nodes and their ranking considerably.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053.

References

1. Domingos, P.: Mining social networks for viral marketing. IEEE Intell. Syst. 20, 80–82 (2005)
2. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12, 211–223 (2001)
3. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: KDD, pp. 1019–1028 (2010)
4. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations 6, 43–52 (2004)
5. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD, pp. 137–146 (2003)
6. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: AAAI 2008, pp. 1175–1180 (2008)
7. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Trans. Knowl. Discov. Data 3, 9:1–9:23 (2009)
8. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: AAAI 2007, pp. 1371–1376 (2007)
9. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Min. and Knowl. Disc. 20, 70–97 (2010)
10. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Learning to predict opinion share in social networks. In: AAAI 2010, pp. 1364–1370 (2010)
11. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 217–226. Springer, Heidelberg (2004)
12. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: EC 2006, pp. 228–237 (2006)
13. Myers, S.A., Leskovec, J.: On the convexity of latent social network inference. In: Proceedings of Neural Information Processing Systems (NIPS)
14. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. 45, 167–256 (2003)
15. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Phys. Rev. E 66, 035101 (2002)
16. Saito, K., Kimura, M., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: SBP 2009, pp. 138–145 (2009)
17. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Zhou, Z.-H., Washio, T. (eds.) ACML 2009. LNCS, vol. 5828, pp. 322–337. Springer, Heidelberg (2009)
18. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Behavioral analyses of information diffusion models by observed data of social network. In: Chai, S.-K., Salerno, J.J., Mabry, P.L. (eds.) SBP 2010. LNCS, vol. 6007, pp. 149–158. Springer, Heidelberg (2010)
19. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Selecting information diffusion models over social networks for behavioral analysis. In: ECML PKDD 2010, pp. 180–195 (2010)
20. Tong, H., Prakash, B.A., Tsoourakakis, C., Eliassi-Rad, T., Faloutsos, C., Chau, D.H.: On the vulnerability of large graphs. In: Perner, P. (ed.) ICDM 2010. LNCS, vol. 6171, pp. 1091–1096. Springer, Heidelberg (2010)
21. Watts, D.J.: A simple model of global cascades on random networks. PNAS 99, 5766–5771 (2002)
22. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. J. Cons. Res. 34, 441–458 (2007)

Analysing the Behaviour of Robot Teams through Relational Sequential Pattern Mining

Grazia Bombini[1], Raquel Ros[2], Stefano Ferilli[1], and Ramon López de Mántaras[3]

[1] University of Bari "Aldo Moro", Department of Computer Science, 70125 Bari, Italy
{gbombini,ferilli}@di.uniba.it
[2] Department of Electrical and Electronic Engineering, Imperial College, UK
r.ros-espinoza@imperial.ac.uk
[3] IIIA - Artificial Intelligence Research Institute, CSIC - Spanish Council for Scientific Research, Campus UAB, 08193 Bellaterra, Spain
mantaras@iiia.csic.es

Abstract. This paper outlines the use of a relational representation in a Multi-Agent domain to model the behaviour of the whole system. The aim of this work is to define a general systematic method to verify the effective collaboration among the members of a team and to compare the different multi-agent behaviours, using external observations of a Multi-Agent System. Observing and analysing the behavior of a such system is a difficult task. Our approach allows to learn sequential behaviours from raw multi-agent observations of a dynamic, complex environment, represented by a set of sequences expressed in first-order logic. In order to discover the underlying knowledge to characterise team behaviours, we propose to use a relational learning algorithm to mine meaningful frequent patterns among the relational sequences. We compared the performance of two soccer teams in a simulated environment, each based on very different behavioural approaches: While one uses a more deliberative strategy, the other one uses a pure reactive one.

1 Introduction

In general in multi-agent domains, and robot soccer in particular, collaboration is desired so that the group of agents work together to achieve a common goal. It is not only important to have the agents collaborate, but also to do it in a coordinated manner so that the task can be organised to obtain effective results. In this work we address the problem of identification of collaborative behaviour in a Multi-Agent System (MAS) environment. The aim is to define a systematic method to verify the effective collaboration among the members of a team and compare the different multi-agent behaviours. Analysing, modelling and recognising agent behaviour external MAS's observations could be very useful to direct team actions. In the analysis of such systems we have dealt with the complexity of the world (continuous and dynamic) state representation and with the recognition of the agent activities. To characterise the state space, it is necessary to represent temporal and spatial state changes.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 163–169, 2011.

A relational representation of team behaviours enables humans to understand and study the action's descriptions of the observed multi-agent systems and the underlying behavioural principles related to the complex changes of state space. Multi-Agent Systems are complex systems made up of several autonomous agent that act to solve different goals. Observing and analysing the behavior of a such system is a difficult task. A relational sequence could be used as a qualitative representation of a team behaviour. This paper addresses the problem of learning and symbolically representing the sequences of actions performed by teams of soccer players, starting from infer the action of a single agent. Low-level concepts of behaviour (events) are recognised and then used to defined high-level concepts (actions). Our proposal is to learn from raw multi-agent observations (log files) of a dynamic and complex environment, a set of relational sequences describing the team behaviour. The method is able to discover strategic events and through the temporal relations between them, to learn interesting actions. The use of relational representations in this context offers many advantages. One of these is generalization across objects and positions. The set of the relational sequences has been used to mine frequent patterns. We use a method based on relational pattern mining to extract meaningful frequent patterns able to define a behavioural team model. A relational sequence is represented by a set of logical atoms. A dimensional atom explicitly refers to dimensional relations between events involved in the sequence. A non-dimensional atom denotes relations between objects, or characterizes an object involved in the sequence. In order to mine frequent patterns, we use an Inductive Logic Programming (ILP) [1] algorithm, based on [2], for discovering relational patterns from sequences. This reduced set represents the common sequences of actions performed by the team and the characteristic behaviour of a team.

2 Learning Behavioural Relational Representation

This section provides a description of the approach that we use to learn relational sequences from log files, which are able to describe and characterise the behaviour of a team of agents. The domain used in this work corresponds to the soccer game, where each team tries to win by kicking a ball into the other team's goal.

The log used represents a stream of consecutive *raw observations* about each soccer player's position and the position of the ball at each time step. From this log streams it is possible to recognise basic actions (*high-level concepts*). Each team has sequences of basic actions used to form coordinated activities which attempt to achieve the team's goals. In our work, we identify the following basic actions of the players:

- **getball**$(T, Player_n)$: at time T, $Player_n$ gains possession of the ball;
- **catch**$(T, Player_n)$: at time T, $Player_n$ gains possession of the ball previously belonging to an opponent;
- **pass**$(T, Player_n, Player_m)$: $Player_n$ kicks the ball and at time T the $Player_m$ gains possession, where both players are from the same team;

- **dribbling**$(T, Player_n)$: at time T, $Player_n$ moves a significant distance avoiding an opponent;
- **progressToGoal**$(T, Player_n)$: at time T, $Player_n$ moves with the ball toward the the penalty box;
- **aloneProgressToGoal**$(T, Player_n)$: at time T, $Player_n$ moves alone with the ball toward the penalty box, without any teammate between it and the goal area;
- **intercept**$(T, Player_n)$: at time T, $Player_n$ loses the possession of the ball, and the new owner of the ball is from the opponent team;

The log stream is processed to infer the low-level *events* that occurred during a trial. An event takes place when the ball possession changes or the ball is out of bounds. A set of recognised events contributes to define an action. Each recognised event has some persistence over time and remains active until another event incompatible with it occurs. An event that occurs in parallel with another event is called a *contemporary* event. It holds until one of the players is able to take full possession of the ball, (i.e. moves away with the ball) or when the ball goes out of bounds. To better describe the behaviour of an entire team, it is necessary to take into account the state of the world and the time in which the action is performed. Agents in dynamic environments have to deal with world representations that change over time. A qualitative description of the world allows a concise and powerful representation of the relevant information. The current world state is represented by the positions of the players (teammates and opponents), and the ball. In this context, to adequately characterise specific scenes, we considered the viewpoint of the player that performs the action to determine how it interacts with others.

Sequences represent a symbolic abstraction of the raw observation. In particular, to describe the relation *direction_view* of the player with respect to the opponent's penalty box, we use $front$, $left$, $right$, *backwards*. To describe the relation of a player with respect to the teammates, the ball and the opponents, we have used two arguments, one for the "horizontal" relation ($forward$ or *behind*) and the other for the "vertical" relation ($left$ or $right$). We use *same* when the player has the same position with respect to the teammate, the ball and the opponents. The following predicates are used the palyer's position:

- **direction_view**$(T, Player_n, position)$;
- **rel_with_ball**$(T, Player_n, horizontal, vertical)$;
- **rel_with_team**$(T, Player_n, horizontal, vertical)$;
- **rel_with_opp1**$(T, Player_n, horizontal, vertical)$;
- **rel_with_opp2**$(T, Player_n, horizontal, vertical)$;

Finally, the following predicates describe the result of the trial:

- **goal**(T): at time T the ball enters into the opponent's goal.
- **to_goal**(T): at time T the ball goes out of the field but passes near one of the goal posts.
- **ball_out**(T): at time T the ball goes out of the field without being a goal or close to goal.
- **block**(T): at time T the goalie stops or kicks the ball.
- **out_of_time**(T): time out.

3 Experimental Evaluation

The aim of this experimentation is to measure and demonstrate the degree of collaboration of soccer teams and, from a more general point of view, to characterise a Multi-Agent System behaviour. Through the pattern mining method, the most frequent set of behaviours is extracted. Two teams of soccer using different behavioural approaches have been analysed. On the one hand, one team follows a strategy based on a Case-Based Reasoning (CBR) approach [3]. The approach allows the players to apply a more deliberative strategy, where they can reason about the state of the game in a more global way, as well as to take into account the opponents when playing. It also includes an explicit coordination mechanism that allows the team to know when and how to act Henceforward we will refer to it as the *CBR* team. On the other hand, the second team follows a reactive approach. An implicit coordination mechanism is defined to avoid having two players "fighting" for the ball at the same time. The resulting behaviour of this approach is more individualistic and reactive. Although they try to avoid opponents (turning before kicking, or dribbling), they do not perform explicit passes between teammates and in general they move with the ball individually. Henceforward we will refer to this approach as the *REA* team.

As we will see, the experiments performed reveal that the action sequences obtained with the approach proposed in this work characterise the behaviour of the *CBR* team as a collaborative team. To be more precise, these action patterns are in the set of most significant patterns extracted from the *CBR* team sequences, whereas they are not among the most significant patterns extracted from the reactive (*REA*) team sequences.

Two sets of simulated[1] experiments, one with the *CBR* team and another one with the *REA* team, were performed. Besides, two possible configurations for the opponents are defined. The first is called DG configuration and considers a defender and a goalie. The second one, the 2D configuration, correspond to a midfield defender and a defender. Four basic scenarios have been defined, each scenario is used with both configurations of opponents (DG or 2D).

In order to evaluate our approach we analyse the recorded observations (log files from the simulated soccer games). We performed 500 trials for each approach (*CBR* and *REA*) and each scenario in the DG configuration, for a total of 4000 trials. The dataset corresponding to these configurations is composed of 10261 sequences (6242 sequences from the *CBR* approach and 4019, from the *REA* approach). Regarding the 2D configuration, we observed that the time required to end a trial was too long. This was due to the ability of the two defenders in preventing the attackers to reach the goal. For this reason a timeout of 60 seconds to end the trial was adopted. For the 2D configuration, we have performed 200 trials per scenario and per approach, obtaining a total of 1600 trials. The dataset is composed of 4329 sequences(2392 sequences for the *CBR* approach and 1937, for the *REA* approach).

[1] The experiments have been run in an extended version of the PuppySim 2 simulator [3], based on the Four-Legged League in RoboCup.

Table 1. Recognised high-level concepts on DG configuration and 2D configuration

		scenario A		scenario B		scenario C		scenario D	
		CBR	REA	CBR	REA	CBR	REA	CBR	REA
DG	N. sequences	1595	977	1513	1170	1623	837	1511	1035
	pass	2285	1325	3135	557	2293	47	2023	190
	dribbling	256	217	254	234	242	161	334	282
	catch	5	3	24	3	10	0	7	0
	intercept	1385	857	1306	1003	1434	771	1324	840
	aloneProgressToGoal	216	35	291	44	192	34	177	34
	progressToGoal	1261	585	974	1200	981	662	511	481
	getball	2583	1246	2467	1423	2461	947	2272	1358
	tot. Actions	7991	4268	8451	4464	7613	2622	6648	3185
2D	N. sequences	622	449	598	477	570	543	602	468
	pass	570	107	865	199	613	95	769	92
	dribbling	77	83	85	73	84	65	63	99
	catch	4	0	6	0	4	0	3	2
	intercept	468	373	460	389	424	468	467	346
	aloneProgressToGoal	34	9	24	17	34	13	40	22
	progressToGoal	342	191	410	350	352	225	459	38
	getball	883	508	907	605	801	682	896	524
	tot. Actions	2378	1271	2757	1633	2312	1548	2697	1123

Table 1 lists the number of sequences that describe the behaviour of the teams. As we can observe the number of sequences for the *CBR* approach is significantly higher than the one used by the reactive approach. Since the *CBR* team plays using collaborative strategies, where the players usually try to reach the goal area by passing the ball to a teammate, or moving to adapted positions to reuse the selected case, more sequences and therefore more actions are needed to describe such behaviour.

When the player holding the ball tries to move towards the penalty area while having in front an opponent, it can act in a cooperative or individualistic way. That is, it can pass the ball to its teammate (in this case the recognised actions would be *getball* and *pass*) or could simply try to overpass the opponent, adopting an individualistic behaviour (if the player succeeds in its aim, the action is recognised as *dribbling*). The number of *pass* actions in the *CBR* sequences is significantly higher than in the ones in the *REA* sequences. On the contrary, the number of the *dribbling* actions within the *REA* sequences is higher than the ones in the *CBR* sequences.

We consider the sequence analysis taking into account only the actions performed during the trials, without considering the predicates describing the state of the world. The goal of this experimentation was to find a subgroup of most meaningful patterns of actions able to characterise the behaviour of a team. We have used the whole dataset, all the sequences of the all scenarios per configuration. Since patterns of low support have a limited coverage of the dataset, these have a very limited discriminative power. But on the other hand, patterns of very high support have also a very limited discriminative power, since they

Table 2. Some interesting patterns

pattern	Fisher score	team
getball(A,B),next_a(A,C),pass(C,B,D)	0.23494427	cbr
pass(A,B,C),next_a(A,D),getball(D,C)	0.07889064	cbr
progressToGoal(A,B),next_a(A,C),pass(C,B,D)	0.03788948	cbr
progressToGoal(A,B),next_a(A,C),intercept(C,B)	0.07138557	rea
progressToGoal(A,B),next_a(A,C),dribbling(C,B)	0.03037611	rea
getball(A,B),next_a(A,C),pass(C,B,D), next_a(C,E), intercept(E,D)	0.05987475	cbr
getball(A,B),next_a(A,C),getball(C,B), next_a(C,D),pass(D,B,E)	0.05517990	cbr
getball(A,B),next_a(A,C),pass(C,B,D), next_a(C,E),progressToGoal(E,D)	0.04290462	cbr
progressToGoal(A,B),next_a(A,C),progressToGoal(C,B), next_a(C,D),intercept(D,B)	0.03860653	rea
progressToGoal(A,B),next_a(A,C),progressToGoal(C,B), next_a(C,D),progressToGoal(D,B)	0.01199806	rea
getball(A,B),next_a(A,C),pass(C,B,D),next_a(C,E), progressToGoal(E,D),next_a(E,F),pass(F,D,B)	0.02503889	cbr
getball(A,B),next_a(A,C),pass(C,B,D),next_a(C,E), getball(E,D),next_a(E,F),pass(F,D,B)	0.01994761	cbr
getball(A,B),next_a(A,C),getball(C,B),next_a(C,D), pass(D,B,E),next_a(D,F),intercept(F,E)	0.01829672	cbr

are too common in the data. Therefore, in general it is appropriate to find not too frequent patterns with suitable support threshold. But this implies a greater effort during the pattern mining step. Frequent patterns reflect strong association between objects, representing common behaviours adopted by a team. The frequency is calculated over the whole dataset and over both sets of sequences (*CBR* and *REA*). Among the different sequences for both teams, the most frequent patterns belong to the *CBR* team. We have used the threshold $\sigma = 0.10$, which is high enough to ensure adequate coverage of the dataset and sufficiently low to allow to discover frequent sequences also for the *REA* team. To select the most meaningful patterns, i.e. a subset of frequent patterns that is able to characterise the essential behaviour of a team, we have used as measure the Fisher Score [4]. It is popularly used in classification system to measure the discriminative power of a feature. Table 2 shows the most interesting patterns obtained by our approach. As we can easily see, the presence of the predicate *pass* is enough to distinguish the *CBR* team. Indeed, this type of action indicates collaborative behaviour, and is typical in sequences that characterise the *CBR* team.

4 Related Work and Conclusions

Some previous work, such as Kaminka et al. [5], focus on unsupervised autonomous learning of the sequential behaviours of agents based on observations of their behaviour. This system identifies and extracts sequences of coordinated

team behaviours from the recorded observations. Similarly to the previous approach, Riley and Veloso [6] model high-level adversarial behaviour by classifying the current opponent team into predefined adversary classes. An observation occurs over a fixed length of time (i.e., window) and it affects the accuracy of the classifier and its performance. Lattner et al. [7] use a sequential pattern mining approach. The process creates patterns in dynamic scenes based on the qualitative information of the environment, and produces a set of prediction rules.

The main difference with respect to the previous work and our approach is the representational power of the learned patterns. Through a logical language it is possible to represent any relations of a complex domain, such as multi-agent system. Furthermore, mined relational patterns are able to represent general characteristics of the teams' behaviours. In this paper we have shown the potential use of a relational representation in a Multi-Agent domain to model the behaviour of the whole system. In this way it is possible to define a high-level description of the multi-agent system's behaviour using multi-agent activity logs. The aim was also to try to measure and demonstrate the degree of collaboration, analysing the joint behaviour of the teams. Starting from infer the action of a single agent, it was possible understand the behavior of the whole system. Low-level concepts of behaviour (events) are recognised and then used to defined high-level concepts (actions). We compared the performance of two soccer teams (*REA* and *CBR*), which have a very different behavioural approach. The results obtained in the experiments confirmed that the recognised action sequences characterise the behaviour of the two teams.

References

1. Muggleton, S., De Raedt, L.: Inductive logic programming: Theory and methods. Journal of Logic Programming 19/20, 629–679 (1994)
2. Esposito, F., Di Mauro, N., Basile, T.M.A., Ferilli, S.: Multi-dimensional relational sequence mining. Fundamenta Informaticae 89(1), 23–43 (2008)
3. Ros, R., Arcos, J.L., de MAntaras, R.L., Veloso, M.M.: A case-based approach for coordinated action selection in robot soccer. Artificial Intelligence 173(9-10), 1014–1039 (2009)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley Interscience, Hoboken (2000)
5. Kaminka, G.A., Fidanboylu, M., Chang, A., Veloso, M.M.: Learning the sequential coordinated behavior of teams from observations. In: Kaminka, G.A., Lima, P.U., Rojas, R. (eds.) RoboCup 2002. LNCS (LNAI), vol. 2752, pp. 111–125. Springer, Heidelberg (2003)
6. Riley, P., Veloso, M.: On behavior classification in adversarial environments. In: Parker, L.E., Bekey, G., Barhen, J. (eds.) Distributed Autonomous Robotic Systems, vol. 4, pp. 371–380. Springer, Heidelberg (2000)
7. Lattner, A., Miene, A., Visser, U., Herzog, O.: Sequential pattern mining for situation and behavior prediction in simulated robotic soccer. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 118–129. Springer, Heidelberg (2006)

Deliberation Dialogues during Multi-agent Planning

Barbara Dunin-Kęplicz[1], Alina Strachocka[2], and Rineke Verbrugge[3]

[1] Institute of Informatics, Warsaw University, Warsaw, Poland,
and ICS, Polish Academy of Sciences, Warsaw, Poland
keplicz@mimuw.edu.pl
[2] Institute of Informatics, Warsaw University, Warsaw, Poland
astrachocka@mimuw.edu.pl
[3] Department of Artificial Intelligence, University of Groningen,
Groningen, The Netherlands
rineke@ai.rug.nl

Abstract. Cooperation in multi-agent systems essentially hinges on appropriate communication. This paper shows how to model communication in teamwork within TeamLog, the first multi-modal framework wholly capturing a methodology for working together. Starting from the dialogue theory of Walton and Krabbe, the paper focuses on deliberation, the main type of dialogue during team planning. We provide a schema of deliberation dialogue along with semantics of adequate speech acts, this way filling the gap in logical modeling of communication during planning.

1 Introduction

Typically teamwork in multi-agent systems (MAS) is studied in the context of BGI (*Beliefs, Goals* and *Intentions*, commonly called BDI) systems, allowing extensive reasoning about agents' informational and motivational attitudes necessary to work together. Along this line, TeamLog [5], a framework for modeling teamwork, has been created on the basis of multi-modal logic. It provides rules for establishing and maintaining a cooperative team of agents, tightly bound by a collective intention and working together on the basis of collective commitment.

Although communication schemes during teamwork were formulated as an inherent part of TeamLog [4], this aspect of multi-agent planning was not yet treated in detail. To fill the gap, a model of deliberation dialogue during planning is investigated in this research. When a team collectively intends to achieve a goal, it needs to decide how to divide this into subgoals, to choose a sequence of actions realizing them, and finally to allocate the actions to team members. We structure these phases as deliberation dialogues, accompanied by ongoing belief revision. Thus, we formally model the team's important transition from a collective intention, to a plan-based social commitment, making it ready for action.

The paper is organized as follows. Sections 2 and 3 briefly introduce speech acts, dialogue, and teamwork theory. Next, in Section 4 the logical language is given, followed by discussion of the consequences of speech acts in Section 5.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 170–181, 2011.

Sections 6 and 7, the heart of the paper, introduce a new model of deliberation and elaborate on planning. Finally, conclusions and plans for future work are presented.

2 Speech Acts and Dialogues

Communication in MAS has two pillars: Walton and Krabbe's semi-formal theory of dialogue [16] and the speech acts theory of Austin and Searle [15,3]. Walton and Krabbe identified six elementary types of dialogues: *persuasion*, *negotiation*, *inquiry*, *information seeking*, *eristics* and, central to this paper, *deliberation*.

Deliberation starts from an open, practical problem: a need for action. It is often viewed as agents' *collective* practical reasoning, where they determine which goals to attend and which actions to perform. While dialogues can be seen as the building blocks of communication, they in turn are constructed from *speech acts*.

Research on speech acts belongs to philosophy of language and linguistics since the early 20th century. The basic observation of Austin [3], that some utterances cannot be verified as true or false, led to the division of speech acts into *constatives*, which can be assigned a logical truth value, and the remaining group of *performatives*. The second father of speech acts theory, Searle, created their most popular taxonomy, identifying: *assertives*, committing to the truth of a proposition (e.g., stating), *directives*, which get the hearer to do something (e.g., asking), *commissives*, committing the speaker to some future action (e.g., promising), *expressives*, expressing a psychological state (e.g., thanking), and *declaratives*, which change reality according to the proposition (e.g., baptising).

Speech acts theory has been extensively used in modeling communication in MAS to express intentions of the sender [8]. There have been many approaches to defining their semantics [13,2,12,9], still some researchers view them as primitive notions [14]. Within the most popular *mentalistic* approach, reflected in languages such as KQML and FIPA ACL [8], speech acts are defined through their impact on agents' mental attitudes. The current paper clearly falls therein (see especially Section 5). Let us place dialogues in the context of teamwork.

3 Stages of Teamwork

In multi-agent cooperative scenarios, communication is inevitable and teamwork, as the pinnacle of cooperation, plays a vital role. The common division of teamwork into four stages originates from [17], while a complete model, binding these stages to formalized team attitudes, can be found in [5]. In summary:

1. Potential recognition. Teamwork begins when an initiator needs assistance and looks for potential groups of agents willing to cooperate to achieve a goal.
2. During **team formation** a loosely-coupled group of agents is transformed into a strictly cooperative team sharing a *collective intention* towards the goal (φ).
3. During **plan formation** a team *deliberates* together how to proceed, concluding in a collective commitment, based on a *social plan*. Collective planning consists of the three phases: *task division*, leading to *division*(φ, σ) (see table 1); *means-end*

analysis, leading to $means(\sigma, \tau)$, and *action allocation*, leading to $allocation(\tau, P)$. Success of these phases is summed up by $constitute(\varphi, P)$.

4. During **team action** agents execute their share of the plan. In real situations, many actions are at risk of failure, calling for a necessary reconfiguration [5], that amounts to the intelligent and situation-sensitive replanning.

With each stage of teamwork, adequate notions in TeamLog are connected. As there is no room for discussing them in detail, please see [5].

Table 1. Formulas and their intended meaning

$\mathrm{BEL}(i, \varphi)$	agent i believes that φ
$\text{E-BEL}_G(\varphi)$	all agents in group G believe φ
$\text{C-BEL}_G(\varphi)$	group G has the common belief that φ
$\mathrm{GOAL}(a, \varphi)$	agent a has the goal to achieve φ
$\mathrm{INT}(a, \varphi)$	agent a has the intention to achieve φ
$\mathrm{COMM}(i, j, \alpha)$	agent i commits to j to perform α
$do\text{-}ac(i, \alpha)$	agent i is just about to perform action α
$division(\varphi, \sigma)$	σ is the sequence of subgoals resulting from decomposition of φ
$means(\sigma, \tau)$	τ is the sequence of actions resulting from means-end analysis on σ
$allocation(\tau, P)$	P is a social plan resulting from allocating the actions from τ
$constitute(\varphi, P)$	P is a correct social plan for achieving φ
$\texttt{confirm}(\varphi)$	plan to test if φ holds at the given world
$\mathrm{prefer}(i, x, y)$	agent i prefers x to y

4 The Logical Language

We introduce a subsystem of $\text{TeamLog}^{\text{dyn}}$ (see [5, Chapters 5 and 6]), containing solely the elements crucial to team planning. Individual actions and formulas are defined inductively.

Definition 1. The language is based on the following sets:
- a denumerable set $\mathcal{P}$ of *propositional symbols*;
- a finite set $\mathcal{A}$ of *agents*, denoted by $1, 2, \ldots, n$;
- a finite set $\mathcal{A}t$ of *atomic actions*, denoted by a or b.

In TeamLog most modalities expressing agents' motivational attitudes appear in two forms: with respect to *propositions* reflecting a particular state of affairs, or with respect to *actions*. The set of formulas $\mathcal{L}$ (see Definition 4) is defined by a simultaneous induction, together with the set of individual actions $\mathcal{A}c$ and the set of social plan expressions $\mathcal{S}p$ (see Definitions 2 and 3). Individual actions may be combined into group actions by the social plan expressions.

Definition 2. The set $\mathcal{A}c$ of individual actions is defined inductively as follows:

AC1 each atomic action $a \in \mathcal{A}t$ is an individual action;
AC2 if $\varphi \in \mathcal{L}$, then $\texttt{confirm}(\varphi)$ is an individual action[1];

[1] In PDL, $\texttt{confirm}(\varphi)$ is usually denoted as "φ?", standing for "proceed if φ is true, else fail".

AC3 if $\alpha_1, \alpha_2 \in \mathcal{A}c$, then $\alpha_1; \alpha_2$ is an individual action, standing for α_1 followed by α_2;
AC4 if $\alpha_1, \alpha_2 \in \mathcal{A}c$, then $\alpha_1 \cup \alpha_2$ is an individual action, standing for nondeterministic choice between α_1 and α_2;
AC5 if $\alpha \in \mathcal{A}c$, then α^* is an individual action, standing for "repeat α a finite, but nondeterministically determined, number of times";
AC6 if $\varphi \in \mathcal{L}$, $i, j \in \mathcal{A}$ and $G \subseteq \mathcal{A}$, then the following are individual actions: $\texttt{announce}_{i,G}(\varphi)$, $\texttt{assert}_{i,j}(\varphi)$, $\texttt{request}_{i,j}(\varphi)$, $\texttt{concede}_{i,j}(\varphi)$.

In addition to the standard dynamic operators of [**AC1**] to [**AC5**], the communicative actions of [**AC6**] are introduced. For their meanings, see Section 5. Their interplay is the main matter of this paper.

Definition 3. The set $\mathcal{S}p$ of social plan expressions is defined inductively:

SP1 If $\alpha \in \mathcal{A}c$, $i \in \mathcal{A}$, then *do-ac*(i, α) is a well-formed social plan expression;
SP2 If $\varphi \in \mathcal{L}$, then $\texttt{confirm}(\varphi)$ is a social plan expression;
SP3 If α and β are social plan expressions, then $\langle \alpha; \beta \rangle$ (sequential composition) and $\langle \alpha \parallel \beta \rangle$ (parallellism) are social plan expressions.

Definition 4. The set of formulas $\mathcal{L}$ is defined inductively:

F1 each atomic proposition $p \in \mathcal{P}$ is a formula;
F2 if $\varphi, \psi \in \mathcal{L}$, then so are $\neg\varphi$ and $\varphi \wedge \psi$;
F3 if $\varphi \in \mathcal{L}$, $\alpha \in \mathcal{A}c$, $i, j \in \mathcal{A}$, $G \subseteq \mathcal{A}$, $\sigma, \sigma_1, \sigma_2$ are finite sequences of formulas, τ is a finite sequence of individual actions, and $P \in \mathcal{S}p$ is a social plan expression, then the following are formulas:
epistemic modalities BEL(i, φ), E-BEL$_G(\varphi)$, C-BEL$_G(\varphi)$;
motivational modalities GOAL(i, φ), INT(i, φ), E-INT$_G(\varphi)$, M-INT$_G(\varphi)$, C-INT$_G(\varphi)$, COMM(i, j, α), S-COMM$_{G,P}(\varphi)$;
execution modalities *do-ac*(i, α);
stage results *division*(φ, σ), *means*(σ, τ), *allocation*(τ, P), *constitute*(φ, P);
other PROOF(φ), prefer(i, σ_1, σ_2).
Epistemic and motivational modalities are governed by the axioms given in the Appendix.

The predicate *constitute*(φ, P) stands for "P is a correctly constructed social plan to achieve φ". Formally:

$$\textit{constitute}(\varphi, P) \leftrightarrow \bigvee_{\sigma} \bigvee_{\tau} (\textit{division}(\varphi, \sigma) \wedge \textit{means}(\sigma, \tau) \wedge \textit{allocation}(\tau, P))$$

A team G has a *mutual intention* to achieve goal φ (M-INT$_G(\varphi)$) if all intend it, all intend that all intend it, and so on, ad infinitum. To create a *collective intention* (C-INT$_G(\varphi)$), a common belief about the mutual intention should be established during team formation. Then, during plan formation, the team chooses a social plan P to achieve φ. On its basis, they create a *collective commitment* (S-COMM$_{G,P}(\varphi)$), including team members' social commitments (COMM(i, j, α)) to perform their allocated actions. The axiom system providing definitions for these notions can be found in the Appendix.

4.1 Kripke Models

Each Kripke model for the language defined above consists of a set of worlds, a set of accessibility relations between worlds, and a valuation of the propositional atoms. The definition also includes semantics for derived operators corresponding to performance of individual actions.

Definition 5. A Kripke model is a tuple:

$$\mathcal{M} = (W, \{B_i : i \in \mathcal{A}\}, \{G_i : i \in \mathcal{A}\}, \{I_i : i \in \mathcal{A}\}, \{R_{i,\alpha} : i \in \mathcal{A}, \alpha \in \mathcal{A}c\}, \mathit{Val}, \mathit{nextac}),$$

such that

1. W is a set of possible worlds, or states;
2. For all $i \in \mathcal{A}$, it holds that $B_i, G_i, I_i \subseteq W \times W$. They stand for the accessibility relations for each agent w.r.t. beliefs, goals, and intentions, respectively.
3. For all $i \in \mathcal{A}$, $\alpha \in \mathcal{A}c$, it holds that $R_{i,\alpha} \subseteq W \times W$. They stand for the dynamic accessibility relations. Here, $(w_1, w_2) \in R_{i,\alpha}$ means that w_2 is a possible resulting state from w_1 by i executing action α.
4. $\mathit{Val} : \mathcal{P} \times W \rightarrow \{0, 1\}$ is the function that assigns the truth values to propositional formulas in states.
5. $\mathit{nextac} : \mathcal{A} \times \mathcal{A}c \rightarrow (W \rightarrow \{0, 1\})$ is the next moment individual action function such that $\mathit{nextac}(i, \alpha)(w)$ indicates that in world w agent i will next perform action α. $\mathcal{M}, v \models \mathit{do\text{-}ac}(i, \alpha) \Leftrightarrow \mathit{nextac}(i, \alpha)(v) = 1$.

In the semantics, the relations $R_{i,a}$ for atomic actions a are given. The other accessibility relations $R_{i,\alpha}$ for actions are built up from these in the usual way [10]:

$$(v, w) \in R_{i,\texttt{confirm}(\varphi)} \Leftrightarrow (v = w \text{ and } \mathcal{M}, v \models \varphi);$$
$$(v, w) \in R_{i,\alpha_1;\alpha_2} \Leftrightarrow \exists u \in W[(v, u) \in R_{i,\alpha_1} \text{ and } (u, w) \in R_{i,\alpha_2}];$$
$$(v, w) \in R_{i,\alpha_1 \cup \alpha_2} \Leftrightarrow [(v, w) \in R_{i,\alpha_1} \text{ or } (v, w) \in R_{i,\alpha_2}];$$
$$R_{i,\alpha^*} \text{ is the reflexive and transitive closure of } R_{i,\alpha}.$$

Definition 6. Let $\varphi \in \mathcal{L}$, $i \in \mathcal{A}$ and $\alpha \in \mathcal{A}c$.

$$\mathcal{M}, v \models [do(i, \alpha)]\varphi \Leftrightarrow \text{ for all } w \text{ with } (v, w) \in R_{i,\alpha}, \mathcal{M}, w \models \varphi.$$

For the dynamic logic of actions, we adapt the axiomatization of propositional dynamic logic (PDL) [10]. The system described above has an EXPTIME-hard decision problem, just like TeamLog$^{\text{dyn}}$ and TeamLog itself [7].

5 Semantics of Speech Acts

In TeamLog, deliberation is modeled via elementary speech acts `assert`, `concede` and `request`, and the compound speech acts `challenge` and `announce`, defined in terms of PDL and described before in [4]. They are treated as ordinary actions and distinguished by their consequences. Utterances often necessitate participants' belief revision, which may be handled by diverse methods (see [1]).

In the sequel, the construction "**if** φ **then** α **else** β" abbreviates the PDL expression $(\texttt{confirm}(\varphi); \alpha) \cup (\texttt{confirm}(\neg\varphi); \beta)$, and analogously for "**if** φ **then** α", where $\texttt{confirm}(\varphi)$ refers to testing whether φ holds (see [5, Chapter 6]). The construct $[\beta]\varphi$ means that after performing β, φ holds.

Consequences of Assertions $\texttt{assert}_{a,i}(\varphi)$ stands for agent a telling agent i that φ holds. According to the fundamental assumption that agents are as truthful as they can be, each $\texttt{assert}(\varphi)$ obliges the sender to believe in φ.

Definition 7. The consequences of assertions:

CA $[\texttt{assert}_{a,i}(\varphi)]\ (\mathrm{BEL}(i,\varphi) \wedge\ \mathrm{BEL}(i,\mathrm{BEL}(a,\varphi)))$

The recipient has two possibilities to react. Unless having beliefs conflicting with φ, it answers with a $\texttt{concede}_{i,a}$. Otherwise, with a $\texttt{challenge}_{i,a}$:

$$\neg\mathrm{BEL}(i,\neg\varphi) \rightarrow \textit{do-ac}(i,\texttt{concede}_{i,a}(\varphi))$$
$$\mathrm{BEL}(i,\neg\varphi) \rightarrow \textit{do-ac}(i,\texttt{challenge}_{i,a}(\varphi))$$

Consequences of Requests $\texttt{request}_{a,i}(\alpha)$ stands for agent a requesting agent i to perform the action α. The sender, after requesting information about φ (with $\alpha = \texttt{assert}_{i,a}(\varphi)$), must wait for a reply. The receiver i has four options:

1. To ignore a and not answer at all.
2. To state that it is not willing to divulge this information.
3. To state that it does not have enough information about φ:
   $$\texttt{assert}_{i,a}(\neg(BEL(i,\varphi) \wedge \neg BEL(i,\neg\varphi))).$$
4. Either to assert that φ is the case or that it is not:
   $\mathrm{BEL}(i,\varphi) \rightarrow \textit{do-ac}(i,\texttt{assert}_{i,a}(\varphi))$ and $\mathrm{BEL}(i,\neg\varphi) \rightarrow \textit{do-ac}(i,\texttt{assert}_{i,a}(\neg\varphi))$.

The consequences are the same as for proper assertions.

Consequences of Concessions $\texttt{concede}_{a,i}(\varphi)$ stands for agent a's communicating its positive attitude towards φ to i. Concessions are similar to assertions. The only difference is that i can assume that a believes φ in the course of dialogue, but might retract it afterwards.

Definition 8. The consequences of concessions:

CCO $[\texttt{concede}_{a,i}(\varphi)]\mathrm{BEL}(i,\mathrm{BEL}(a,\varphi))$.

Consequences of Challenges $\texttt{challenge}_{a,i}(\varphi)$ stands for a's communicating its negative attitude towards φ to i. The consequences of `challenge` are more complicated due to the complexity of the speech act itself. It consists of a negation of φ and of a request to prove φ.

Definition 9. If $\varphi, PROOF(\varphi) \in \mathcal{L}$, $a, i \in \mathcal{A}$, then

CH $\texttt{challenge}_{a,i}(\varphi) \equiv\ \texttt{assert}_{a,i}(\neg\varphi);\texttt{request}_{a,i}(\texttt{assert}_{i,a}(PROOF(\varphi)))$

The answer to the `request` in `challenge` should comply with the rules above. If i can prove φ, it should answer with speech act $\texttt{assert}_{i,a}(PROOF(\varphi))$ being committed to $PROOF(\varphi)$. In return, a should refer to i's previous answer. Thus[2], the consequences of `challenge` depend on the dialogue and can be twofold.

[2] Assuming the rule $\mathrm{BEL}(a,PROOF(\varphi)) \rightarrow \mathrm{BEL}(a,\varphi)$.

Definition 10. The consequences of challenges:

CH1 $[\texttt{challenge}_{a,i}(\varphi)]$ $(\mathrm{BEL}(a,\varphi) \wedge \mathrm{BEL}(i,\mathrm{BEL}(a,\varphi)) \wedge \mathrm{BEL}(a,PROOF(\varphi)) \wedge \mathrm{BEL}(i,\mathrm{BEL}(a,PROOF(\varphi))))$

CH2 $[\texttt{challenge}_{a,i}(\varphi)]$ $(\neg\mathrm{BEL}(i,\varphi) \wedge \mathrm{BEL}(a,\neg\mathrm{BEL}(i,\varphi)) \wedge \neg\mathrm{BEL}(i,PROOF(\varphi)) \wedge \mathrm{BEL}(a,\neg\mathrm{BEL}(i,PROOF(\varphi))))$

In first case, **[CH1]**, a admits it was wrong. The agents' beliefs have changed, reflected by the acceptance of i's proof, which led to belief revision about φ. In the second case, i admits it was wrong. Belief revision regarding rejecting the proof of φ leads to updating beliefs about φ.

Consequences of Announcements An announcement $\texttt{announce}_{a,G}(\varphi)$ can be seen as a complex assertion standing for "agent a announces to group G that φ holds". In addition, the agent passes a message that the same information has been delivered to the whole group. The group becomes commonly aware that φ.

Definition 11. Consequences of announcements:

CAN $[\texttt{announce}_{a,G}(\varphi)]$ $\text{C-BEL}_G(\varphi)$.

Once the logical language is set, we may proceed to the core of this paper.

6 A Four-Stage Model of Deliberation

The schema for deliberation dialogues presented below benefits from the model of McBurney, Hitchcock and Parsons [11]. It starts from a formal opening, introducing the subject of the dialogue, aiming to make a common decision, confirmed in a formal closure. Deliberation on "$\psi(x)$"[3] aims at finding the best t satisfying ψ from a finite candidate set T_ψ and to create a common belief about this among the team. Even though deliberation during teamwork is a collective activity, its structure is imposed by the initiator a. Other agents follow the rules presented below. Failure at any of the dialogue stages causes backtracking.

Opening. Agent a's first step is to open the deliberation dialogue on the subject ψ by a request to all other $i \in G$:

$$\texttt{request}_{a,i}(\ \textbf{if}\ \bigvee_{t\in T_\psi}\psi(t)\ \textbf{then}\ \texttt{assert}_{i,a}(\psi(t))\textbf{else}\ \texttt{assert}_{i,a}(\neg\bigvee_{t\in T_\psi}\psi(t)))$$

As always after requests, agents have four ways of answering (see Section 5). If no one answers, deliberation fails. Agent a waits for a certain amount of time before concluding on the answers from group G.

Voting. During voting, a announces to all $i \in G$ its finite set $T_{\psi,a}$ of all or preselected answers collected before:

$$\texttt{assert}_{a,i}(\bigwedge_{t\in T_{\psi,a}}\bigvee_{i\in G}\mathrm{BEL}(i,\psi(t)))$$

[3] ψ is usually ungrounded, e.g. $\psi(x) = president(x)$. The answers are (partially) grounded terms, e.g., $president(JohnSmith)$.

In words, a asserts to every agent that for each preselected answer t, there is an agent i in the group believing that $\psi(t)$ is the case.
Next, agent a opens the voting by a request to all $i \in G$:

$$\texttt{request}_{a,i}(\bigwedge_{x,y \in T_{\psi,a}} (\textbf{ if } \text{BEL}(i, \psi(x)) \wedge \text{prefer}(i, x, y) \textbf{ then } \texttt{assert}_{i,a}(\text{prefer}(i, x, y))))$$

If no one answers, the scenario leads back to step 1, which is justified because the communication in step 2 may entail some belief revisions. Should some answers be received, a "counts the votes", possibly using different evaluation functions.

Confirming. Then, a announces the winning proposal w and requests all opponents from G to start a persuasion:

$$\texttt{request}_{a,i}(\textbf{if } \text{BEL}(i, \neg\psi(w)) \vee \bigvee_{t \in T_{\psi,a}} (\text{prefer}(i, t, w)) \textbf{ then } \texttt{assert}_{i,a}(\neg\psi(w) \vee \text{prefer}(i, t, w)))$$

During this phase, if no agent steps out, the scenario moves to the closure. If, on the other hand, there is an agent j who thinks that w is not the best option, it has to announce this and challenge a to provide a proof (using `challenge`). Thus the dialogue switches to persuasion, where j must convince a of the competing offer t, or that $\psi(w)$ doesn't hold. If it succeeds, a adopts and heralds agent's j thesis to all $i \in G$:

$$\texttt{assert}_{a,i}(\neg\psi(w) \vee \text{prefer}(a, t, w))$$

In this situation, the remaining agents may concede or may challenge the thesis:

$$\texttt{concede}_{i,a}(\neg\psi(w) \vee \text{prefer}(a, t, w)) \text{ or } \texttt{challenge}_{i,a}(\neg\psi(w) \vee \text{prefer}(a, t, w)).$$

If they choose to challenge, a must get involved into persuasion with the challenging agent. Finally, when all conflicts have been resolved, the scenario moves on.

Closure. At last, a announces the final decision w: $\texttt{announce}_{a,G}(\psi(w))$.

Deliberating agents collaborate on the future course of actions, each of them trying to influence the final outcome. The principal kind of reasoning here is goal-directed practical reasoning, leading to a plan.

7 Unveiling the Plan Formation Stage

The (formal) aim of plan formation is transition from a collective intention to a collective commitment (see Section 3), achieved by means of dialogue. Consider, as an example, a team of various robots: digger (D), truck (T) and team leader (L) with a goal to restore order after a building collapses: $\varphi = \texttt{order}$. They discover another working team soon: first aid (FA) and 20 swarm robots ($S_1, \ldots, S_{20}$) and decide to join forces to better serve their goal. Suppose potential recognition and team formation have been successful. Then, the first phase of planning, *task division*, aims at dividing the overall goal φ into a new sequence of subgoals. L opens the deliberation dialogue by requesting all other $i \in G$ to share their ideas:

$$\texttt{request}_{L,i}(\ \textbf{if}\ \bigvee_{\sigma\in T_{Goals}} \mathit{division}(\texttt{order},\sigma)\ \textbf{then}\ \texttt{assert}_{i,L}(\mathit{division}(\texttt{order},\sigma))\ \textbf{else}\ \texttt{assert}_{i,L}(\neg\bigvee_{\sigma\in T_{Goals}} \mathit{division}(\texttt{order},\sigma))),$$

where σ is a sequence of goals from a pre-given finite set of goals T_{Goals}. L waits a while before collecting the answers from G. Suppose two agents decide to respond: D and FA. D proposes the sequence σ_D:

$$\sigma_D = \langle \texttt{scan\_ruins}, \texttt{clear\_safe}, \texttt{clear\_risky}, \texttt{fist\_aid\_risky}\rangle.$$

In other words, first, scan the ruins of the building in search for survivors, then, clear the area with no sign of people, next, clear the area where survivors might be and finally, help survivors. In FA's view, the only difference is the goal order:

$$\sigma_{FA} = \langle \texttt{scan\_ruins}, \texttt{clear\_risky}, \texttt{fist\_aid\_risky}, \texttt{clear\_safe}\rangle.$$

As a response to L's call, the two agents utter:

$$\texttt{assert}_{D,L}(\mathit{division}(\texttt{order},\sigma_D)) \text{ and } \texttt{assert}_{FA,L}(\mathit{division}(\texttt{order},\sigma_{FA}))$$

These two assertions cause belief revision. The second step is voting. The pre-selected subset of answers collected previously (candidate terms) is $T_{\texttt{order},L} = \{\sigma_D, \sigma_{FA}\}$. L discloses this information to all other agents $i \in G$:

$$\texttt{assert}_{L,i}\left(\bigwedge_{t\in T_{\texttt{order},L}} \bigvee_{i\in G} \mathrm{BEL}(i, \mathit{division}(\texttt{order},t))\right)$$

Subsequently, L opens voting on proposals by requests to all $i \in G$:

$$\texttt{request}_{L,i}(\bigwedge_{x,y\in T_{\texttt{order},L}}(\ \textbf{if}\ \ \mathrm{BEL}(i, \mathit{division}(\texttt{order},x)) \wedge \mathrm{prefer}(i,x,y)\ \ \textbf{then}\ \ \texttt{assert}_{i,L}(\mathrm{prefer}(i,x,y))))$$

In step 3 (confirming), L announces that for example σ_{FA} won and calls potential opponents to start a persuasion dialogue, by sending a request to all other $i \in G$:

$$\texttt{request}_{L,i}(\ \textbf{if}\ \mathrm{BEL}(i, \neg \mathit{division}(\texttt{order},\sigma_{FA})) \vee \bigvee_{t\in T_{\texttt{order},L}} \mathrm{prefer}(i,t,\sigma_{FA})\ \textbf{then}\ \texttt{assert}_{i,L}(\neg \mathit{division}(\texttt{order},\sigma_{FA}) \vee \mathrm{prefer}(i,t,\sigma_{FA})))$$

If agent D prefers its own proposal, it raises an objection:

$$\texttt{assert}_{D,L}(\mathrm{prefer}(i,\sigma_D,\sigma_{FA}))$$

This is followed by L's challenge to provide a proof. At this point, the dialogue switches to persuasion, which has been discussed in [4,5]. Step 4 (closure) follows the same pattern, leading, if successful, to a subgoal sequence σ such that $\mathit{division}(\texttt{order},\sigma)$ holds.

The next stage is *means-end-analysis*, when every subgoal must be assigned a (complex) action realizing it. If the whole process concerning all subgoals from σ succeeds, there is an action sequence τ such that $\mathit{means}(\sigma,\tau)$ holds. The last phase, *action allocation*, results, if successful, in a plan P for which $\mathit{allocation}(\tau,P)$ holds. Finally, $\mathit{constitute}(\texttt{order},P)$ is reached and planning terminates. There is now a basis to establish a collective commitment and to start working.

Although deliberation during teamwork is a complex process, all its phases can be naturally specified in TeamLog. First, central to teamwork theory, collective

group attitudes are defined in terms of other informational and motivational attitudes (via fixpoint definitions). Then, the dynamic component allows specifying consequences of various speech acts, plans, and complex actions. These can be further applied as building blocks of different dialogues. The entire multi-modal framework constituting TeamLog is presented in the recent book [5].

8 Conclusions and Future Work

We have introduced a novel approach to modeling deliberation dialogues in teamwork. Although dialogues and speech acts have been frequently used to model communication in multi-agent systems [8], the TeamLog solution is unique. The proposed scenario consists of four stages, during which agents submit their proposals, vote on preferred ones and challenge or concede the choice of the selected one. Depending on the context and aim of the system, a system designer decides on tactical aspects such as waiting time for the leader during answer collection, and manner of counting votes, possibly using weights depending on the agents and/or types of proposals.

The scenario specifies precisely when to embed other dialogues types into deliberation, as opposed to [11]. Different types of dialogues are strictly distinguished, and the boundary between them is clearly outlined, providing an appropriate amount of flexibility, enabling smooth teamwork (about the importance of dialogue embedding, see also [6]). In the course of deliberation, a social plan leading to the overall goal is created, belief revision is done and growth of knowledge can be observed. Finally, along with existing schemas for persuasion and information seeking [4], the most vital aspects of communication in TeamLog are now addressed.

In future, communication involving uncertain and possibly inconsistent information will be investigated, most probably requiring a new model of TeamLog.

Acknowledgements

This research has been supported by the Polish grant N N206 399334 and Dutch Vici grant NWO-277-80-001.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic 50(2), 510–530 (1985)
2. Atkinson, K., Bench-Capon, T., McBurney, P.: Computational representation of practical argument. Synthese 152, 157–206 (2005)
3. Austin, J.L.: How to Do Things with Words, 2nd edn., Urmson, J.O., Sbisa, M. (eds.). Clarendon Press, Oxford (1975)
4. Dignum, F., Dunin-Kęplicz, B., Verbrugge, R.: Creating collective intention through dialogue. Logic Journal of the IGPL 9, 145–158 (2001)

5. Dunin-Kęplicz, B., Verbrugge, R.: Teamwork in Multi-Agent Systems: A Formal Approach. Wiley, Chichester (2010)
6. Dunin-Kęplicz, B., Verbrugge, R.: Dialogue in teamwork. In: Balkema, A.A. (ed.) Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications, Rotterdam, pp. 121–128 (2003)
7. Dziubiński, M., Verbrugge, R., Dunin-Kęplicz, B.: Complexity issues in multiagent logics. Fundamenta Informaticae 75(1-4), 239–262 (2007)
8. FIPA (2002), http://www.fipa.org/
9. Guerin, F., Pitt, J.: Denotational semantics for agent communication language. In: AGENTS 2001: Proceedings of the Fifth International Conference on Autonomous Agents, pp. 497–504. ACM, New York (2001)
10. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)
11. McBurney, P., Hitchcock, D., Parsons, S.: The eightfold way of deliberation dialogue. International Journal of Intelligent Systems 22(1), 95–132 (2007)
12. McBurney, P., Parsons, S.: A denotational semantics for deliberation dialogues. In: AAMAS 2004: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 86–93. IEEE Computer Society, Washington, DC,USA (2004)
13. Parsons, S., McBurney, P.: Argumentation-based dialogues for agent coordination. Group Decision and Negotiation 12, 415–439 (2003)
14. Prakken, H.: Formal systems for persuasion dialogue. The Knowledge Engineering Review 21(2), 163–188 (2006)
15. Searle, J.R., Vanderveken, D.: Foundations of Illocutionary Logic. Cambridge University Press, Cambridge (1985)
16. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany (1995)
17. Wooldridge, M., Jennings, N.R.: The cooperative problem-solving process. Journal of Logic and Computation 9(4), 563–592 (1999)

A Appendix: Axiom Systems

All axiom systems introduced here are based on the finite set $\mathcal{A}$ of n agents.

General Axiom and Rule. The following cover propositional reasoning:

P1. All instances of propositional tautologies;
PR1. From φ and $\varphi \rightarrow \psi$, derive ψ;

Axioms and Rules for Individual Belief, Goal and Intention. For beliefs, the $\mathrm{KD45}_n$ system for n agents is adopted, for intentions, KD_n, for goals, K_n.

Interdependencies Between Intentions and Other Attitudes. For each $i \in \mathcal{A}$:

$\mathbf{A7}_{DB}$ $\mathrm{GOAL}(i,\varphi) \rightarrow \mathrm{BEL}(i,\mathrm{GOAL}(i,\varphi))$
$\mathbf{A7}_{IB}$ $\mathrm{INT}(i,\varphi) \rightarrow \mathrm{BEL}(i,\mathrm{INT}(i,\varphi))$
$\mathbf{A8}_{DB}$ $\neg\mathrm{GOAL}(i,\varphi) \rightarrow \mathrm{BEL}(i,\neg\mathrm{GOAL}(i,\varphi))$
$\mathbf{A8}_{IB}$ $\neg\mathrm{INT}(i,\varphi) \rightarrow \mathrm{BEL}(i,\neg\mathrm{INT}(i,\varphi))$
$\mathbf{A9}_{ID}$ $\mathrm{INT}(i,\varphi) \rightarrow \mathrm{GOAL}(i,\varphi)$

Axioms and Rule For General ("Everyone") and Common Belief

C1 $\text{E-BEL}_G(\varphi) \leftrightarrow \bigwedge_{i \in G} \text{BEL}(i, \varphi)$
C2 $\text{C-BEL}_G(\varphi) \leftrightarrow \text{E-BEL}_G(\varphi \wedge \text{C-BEL}_G(\varphi))$
RC1 From $\varphi \rightarrow \text{E-BEL}_G(\psi \wedge \varphi)$ infer $\varphi \rightarrow \text{C-BEL}_G(\psi)$

Axioms and Rule for Mutual and Collective Intentions

M1 $\text{E-INT}_G(\varphi) \leftrightarrow \bigwedge_{i \in G} \text{INT}(i, \varphi)$
M2 $\text{M-INT}_G(\varphi) \leftrightarrow \text{E-INT}_G(\varphi \wedge \text{M-INT}_G(\varphi))$
M3 $\text{C-INT}_G(\varphi) \leftrightarrow \text{M-INT}_G(\varphi) \wedge \text{C-BEL}_G(\text{M-INT}_G(\varphi))$
RM1 From $\varphi \rightarrow \text{E-INT}_G(\psi \wedge \varphi)$ infer $\varphi \rightarrow \text{M-INT}_G(\psi)$

Defining Axiom for Social Commitment

$$\text{COMM}(i, j, \alpha) \leftrightarrow \text{INT}(i, \alpha) \wedge \text{GOAL}(j, \mathit{done}(i, \alpha)) \wedge$$
$$\mathit{awareness}_{\{i,j\}}(\text{INT}(i, \alpha) \wedge \text{GOAL}(j, \mathit{done}(i, \alpha)))$$

Defining Axiom for Strong Collective Commitment

$$\begin{aligned}&\text{S-COMM}_{G,P}(\varphi) \leftrightarrow \text{C-INT}_G(\varphi) \wedge \\ &\mathit{constitute}(\varphi, P) \wedge \text{C-BEL}_G(\mathit{constitute}(\varphi, P)) \wedge \\ &\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha) \wedge \text{C-BEL}_G(\bigwedge_{\alpha \in P} \bigvee_{i,j \in G} \text{COMM}(i, j, \alpha))\end{aligned}$$

TeamLog denotes the union of the axioms for individual attitudes with the above axioms and rules for general and common beliefs and for general, mutual and collective intentions. TeamLog$^{\text{com}}$ denotes the union of TeamLog with the axioms for social and collective commitments (see [5, Chapter 4]). By TeamLog$^{\text{dyn}}$ we denote the union of TeamLog$^{\text{com}}$ with the axioms for dynamic operators, adopted from [10].

DDLD-Based Reasoning for MAS*

Przemysław Więch, Henryk Rybinski, and Dominik Ryżko

Institute of Computer Science, Warsaw University of Technology
{pwiech,hrb,d.ryzko}@ii.pw.edu.pl

Abstract. In this paper, a model for DDL$\mathfrak{D}$-based multi-agent system is described. The article extends our previous work, in which a formalism for distributed default reasoning to be performed by a group of agents that share knowledge in the form of a distributed default theory has been presented. The formalism is based on default transformations, which can be used to derive answers to queries in the form of defaults. The distributed reasoning process is described in a setting where agents communicate by passing messages.

Keywords: multi-agent system, default logic, description logic, distributed reasoning.

1 Introduction

Many real world applications require knowledge, which is distributed and is not located at the entity assigned to solve the given problem. Such entities must be able to cooperate in order to reach solutions of the problems presented to them. This is actually the approach of multi-agent systems (in the sequel MAS), which provide tools for modelling the situations by means of a set of collaborating autonomous, intelligent and proactive agents. Examples of applications of MAS in the area of the energy markets are shown in [8].

In the Semantic Web, knowledge is distributed throughout the Web and it can be seen as a network of agents, each having its own knowledge base and reasoning facilities. The entities can have specialized knowledge, which can be shared and reused by agents that need to collect remote information in order to perform a reasoning task. Distributed reasoning in a peer-to-peer setting is shown in [1], where a message passing algorithm is introduced to exchange knowledge between peers. Each peer runs an inference procedure on local knowledge to answer queries from neighbouring peers. Here, in contrast to other approaches, the global theory defined as a sum of all local knowledge is unknown.

In [15], we have argued that it is beneficial to enable the agents to exchange information in the form of defaults as these contain additional information about default justifications and thus can prevent the loss of information.

* This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 182–191, 2011.

In this paper, we present a model for DDL$\mathfrak{D}$-based MAS, which utilizes sharing of knowledge among agents in order to achieve its goals. The agents communicate using messages with queries and answers. The model of an agent's knowledge base is introduced and the algorithms for distributed reasoning are presented.

2 Related Work

Logic is often used as the basis for knowledge representation in multi-agent systems. In [9], Kowalski and Sadri describe an extension of logic programming to provide rationality and reactiveness in the multi-agent setting.

In a distributed environment, the knowledge is scattered among the agents. The field of theory partitioning studies the methods of dividing a logical theory in order to increase the efficiency of reasoning. Amir and McIlraith [2] introduce forward and backward reasoning algorithms for a partitioned first-order logic theory. Here, message passing is used to transfer knowledge between partitions.

In [12,13] a multi-agent system is proposed for knowledge sharing in an environment of agents equipped with default reasoning abilities. The Distributed Default Logic framework (DDL) is composed of agents having their knowledge in the form of default logic theories, and able to communicate with each other in order to resolve the locally unknown facts.

Distributed reasoning is essential for the domain of the Semantic Web as the knowledge is inherently distributed among many sources. The Semantic Web bases its knowledge representation on Description Logics (DLs) [3]. On the grounds of the DL formalisms, several approaches to mapping distributed knowledge bases have been investigated [6,7,5]. Our work extends the notions of *Distributed Description Logic* by introducing defaults to the knowledge representation formalism and to the inference procedure.

In the remainder of the paper we refer to Default Logic as defined by Reiter [11].

3 Basic Concepts

Description logics (DLs) [3] are a family of knowledge representation formalisms. Knowledge in DLs is represented by defining concepts from a selected domain, which comprise a terminology, and using these concepts for classifying objects and describing their properties. A DL knowledge base is composed of a terminological part (TBox), where axioms describing relationships between concepts, and an assertional part (ABox), which expresses the inclusion of individuals to specific concepts (e.g. $C(a)$). The DL descriptions are formed by combining concept names with constructors. The basic DL constructors are conjunction ($C \sqcup D$), disjunction ($C \sqcap D$) and complement ($\neg C$), where C and D are concept names.

Baader and Hollunder [4] show how defaults can be embedded into description logics in order to allow commonsense reasoning.

Definition 1. *A default is in the form* $\frac{A:B}{C}$ *where A, B and C are concept expressions. This notation is equivalent to expressing the default in which concepts are expressed as unary predicates* $\frac{A(x):B(x)}{C(x)}$

The default expresses that it can be inferred that x is an instance of the concept C if x is an instance of A and it is consistent to assume that x is an instance of B.

Embedding defaults in DLs is not as straightforward as it may seem. The problem is with treatment of open defaults by Skolemization. A terminological knowledge base with defaults is undecidable, unless we consider only closed defaults. This means that defaults can only be applied to named individuals which already exist in the knowledge base.

A normal default in the form $\frac{A:B}{B}$ can be seen as a weaker form of subsumption, such that it permits exceptions. The default $\frac{A:\top}{B}$ is also weaker than the axiom $A \sqsubseteq B$ because although there is no possibility of specifying exceptions, it does not imply the contrapositive $\neg B \sqsubseteq \neg A$.

4 Distributed Reasoning with Defaults

The main motivation for default transformation is to provide more informative answers in the form of defaults, which can be used with information posessed by the querying agent.

4.1 Transforming Defaults

As described in [15], when using defaults in the reasoning process the answer to an agent's query should also carry the information about assumptions made during the reasoning process. A set of transofrmation rules has been presented, which have the property that when they are added to the default theory, the theory does not change with respect to the results of reasoning. In other words, the set of extensions of the default theory must remain unchanged.

Definition 2. *A* **default transformation** $t : \Delta \rightarrow \mathcal{D}$ *produces a new default* δ *from a default theory* $\Delta = \langle D, W \rangle$ *and is denoted by* $\Delta \mid\sim \delta$.

We define a set of transformations which have very useful features and will be used in the process of distributed reasoning. A general form of a transformation is $\langle D_t, f_t \rangle \mid\sim \delta$, where $D_t \subseteq D$, $W \models f_t$, and δ is a new concluded default.

Definition 3. *Given well-formed formulae* a, b, c, d, e, *we define the following transformations:*

a). Prerequisite substitution: $\left\langle \{\frac{a:b \wedge c}{b}\}, d \rightarrow a \right\rangle \quad \mid\sim \frac{d:b \wedge c}{b}$

b). Consequent substitution: $\left\langle \{\frac{a:b \wedge c}{b}\}, b \rightarrow e \right\rangle \quad \mid\sim \frac{a:b \wedge c \wedge e}{e}$

c). Justification reduction: $\left\langle \{\frac{a:b\sqcap c\sqcap d}{b}\}, a \to d \right\rangle \mid\sim \frac{a:b\sqcap c}{b}$

d). Default transitivity: $\left\langle \{\frac{a:b\sqcap c}{b}, \frac{b:e\sqcap f}{e}\}, \top \right\rangle \mid\sim \frac{a:b\sqcap c\sqcap e\sqcap f}{b\sqcap e}$

The set of transformations (a)–(d) will be called *basic transformations*. These transformations can be further used in the communication process. Theorem 1 shows the interesting property of these transformations.

Let us define the sequence of default transformations, denoted by $\mid\sim_*$, as follows. Let $D_0, \ldots, D_n$ be a sequence such that $D_0 = D$ and $D_i = D_{i-1} \cup \{\delta_i\}$ where δ_i is obtained by applying a basic transformation on $\langle D_{i-1}, W\rangle$. We write $\langle D, W\rangle \mid\sim_* \delta$ when $\langle D_n, W\rangle \mid\sim \delta$.

Theorem 1. *Given* $\Delta = \langle D, W\rangle$ *and* $\Delta' = \langle D', W\rangle$*, where* $\forall_{\delta\in D'}(\delta \in D$ *or* $D \mid\sim_* \delta)$*, we have* $ext(\Delta) = ext(\Delta')$

The theorem shows that using the defined basic transformations we can create new defaults, which can be treated as valid rules for default reasoning. Moreover, these newly formed defaults can be treated as intermediate results of inference. For a full proof of the theorem see [16].

4.2 Reasoning with Default Transformations

In a multi-agent system the peers exchange knowledge by means of querying each other and utilising the answers to reach conclusions. Following the inference procedure for Distributed Description Logic proposed in [14], the query, which is passed between agents is the subsumption query in the form $A \sqsubseteq B$, which in first-order logic can be denoted as $A(x) \to B(x)$. Here, we will concentrate on this type of query and we will denote it by writing $A \sqsubseteq B?$ to distinguish it from a DL statement.

For a query $A \sqsubseteq B?$ to a default theory $\Delta = \langle D, W\rangle$ we will presume there are three possible answers:

- *true* if $W \models A \sqsubseteq B$
- *false* if $W \not\models A \sqsubseteq B$,
- *true by default* if the default $\frac{A\,:\,B\sqcap J}{B}$ can be generated using the default transformations

The first two answers are strict and do not require further processing. The last answer can be treated as a partial result and the final answer can be inferred when the justifications are checked. The algorithm for reasoning with default transformations is described in detail in [15].

The algorithm first checks if a trivial answer can be given without the need to process defaults. If such an answer cannot be given, the next step is to find all extensions of the default theory. Iterating over all extensions, the procedure gathers defaults in the form $\frac{A\,:\,B\sqcap J}{B}$, possibly from different extensions. This is done by transforming the generating defaults of each extension. Finally the resulting defaults are processed, applying the *reduce justifications* transformation.

In effect the query algorithm generates one of three possible answers, which can be *true*, *false* or a set of defaults which are in the form $\frac{A\,:\,B\sqcap J}{B}$.

5 DDL$\mathfrak{D}$-Based Multi-Agent System

In this section we present the model for knowledge representation and exchange in a multi-agent system. The figure below shows an overview of the system. Each agent is an independent entity in the system and contains its own knowledge base and inference engine. Note that the agents may also perform other tasks, however, this work concentrates on the exchange of information in a multi-agent system.

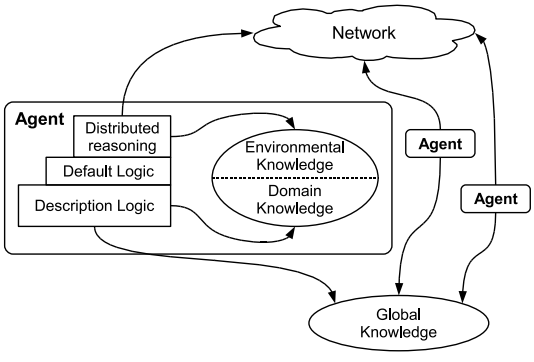


Fig. 1. Overview of the multi-agent system

5.1 Agent Knowledge

Each agent has exclusive direct access to its *local knowledge* (LK). The agent's local knowledge base is divided into two parts. *Domain knowledge* (DK) contains information connected with the agent's primary activity, while *environmental knowledge* (EK) includes information about other agents and information sources. Given the i-th agent in the MAS, the agent's knowledge is expressed as $LK_i = \langle DK_i, EK_i \rangle$. In our approach, domain knowledge does not depend on the environmental knowledge and can be used for local reasoning. Apart from having local knowledge bases, all agents in the multi-agent system share *global knowledge* (GK), which sets the framework for the agent communication language.

The local domain knowledge (DK) and the global ontologies (GK) are expressed in terms of description logic with defaults. The local reasoning processes take into account both of these knowledge bases ($DK \cup GK$). This makes it possible to formulate queries to other agents in terms of the common vocabulary. The component responsible for distributed reasoning utilizes the environmental knowledge in order to find information required for the current reasoning task by communicating with other agents

It is assumed that an agent's local domain knowledge is consistent with the global knowledge. Otherwise, the overall knowledge base of the agent would be inconsistent and would not be useful for reasoning. Moreover, it is assumed

that all concepts, roles and individuals are globally uniquely identified. This means that no two concepts with the same name can exist, which have different semantical meanings associated with them. Globally unique identifiers make information, which is communicated, unambiguous.

The agents in a multi-agent system exchange information using a communication language and a protocol for specifying types of messages and the available replies. For the purpose of distributed reasoning, we will use a communication protocol, where agent X sends a query to agent Y asking if the other peer knows anything about the concept A being subsumed by concept B. We denote this query as $A \sqsubseteq B?$. Although in description logic the axiom $A \sqsubseteq B$ is equivalent to $\neg B \sqsubseteq \neg A$, the query $\neg B \sqsubseteq \neg A?$ is considered as a different query due to the form of possible replies.

Agent Y replies to a query $A \sqsubseteq B?$ with one of the following statements:

- TRUE – Agent Y entails $A \sqsubseteq B$
- FALSE – Agent Y does not entail $A \sqsubseteq B$
- Set of defaults d_i in the form $\frac{A : J_i \sqcap B}{B}$, where J_i are the defaults' justifications.

Definition 4. *The* **vocabulary** V *of a knowledge base* KB *is the set of all concept names occurring in* KB *and is denoted as* $V(KB) \subseteq \mathbb{T}$*, where* $\mathbb{T}$ *is the set of all possible concept names.*

The vocabulary of the i-th agent in the multi-agent system will be denoted as $V_i = V(DK_i \cup GK)$. The *common vocabulary* of two agents i and j is the intersection of the agents' vocabularies $V_i \cap V_j$. Note that the common vocabulary will always contain the vocabulary of the global knowledge $V(GK) \subseteq V_i \cap V_j$ and may also contain additional common concept names, which both agents understand.

For an agent to know with which agents it can communicate about which topics, it needs to possess environmental knowledge about peers it can connect to. Two agents can exchange messages only if both of them share a common vocabulary.

In order to manage the information about its peers, each agent maintains environmental knowledge, which provides information about how to interact with the environment and other agents. The distributed reasoning procedure utilizes both types of knowledge to execute inference tasks in a distributed environment. Environmental knowledge can be expressed as the relation between the set of agents and the set of concepts $EK \subseteq MAS \times \mathbb{T}$, where MAS is the set of agents in the multi-agent system and $\mathbb{T}$ is the set of all possible concept names.

In this work we will assume that the environmental knowledge is given a priori. However, it is a valuable topic to investigate how to acquire such information. The work by Ryżko [12] describes a multi-agent system using explanation based learning to acquire environmental knowledge through learning.

5.2 Distributed Reasoning

The distributed reasoning component of the agent deals with identifying queries to be issued to remote information sources, choosing agents to communicate with and then sending appropriate queries.

The first problem is to identify such queries, for which the answers could change the inferences. There is no need to ask questions, which have no influence on the outcome of the current inference task. The agent uses the tableau method to perform inferences on the local knowledge base. In short, this algorithm tries to create a model of the description logic knowledge base by building a set of ABoxes. A disjunction in the KB creates two branches, which have to be expanded and checked for consistency. The algorithm ends when one of the branches is complete and does not contain an obvious contradiction (a clash) or all branches are closed containing clashes. The outcome of the tableau algorithm can be changed if adding a new piece of information causes an open branch to close.

In consequence, the goal of issuing queries to other agents is to close branches, which would not be closed only basing on the local knowledge base. Suppose the ABox $\mathcal{A}$ contains two assertions $A(a)$ and $B(a)$. In the local KB they do not produce a clash. However, if another source can provide information that $(\neg B \sqcup \neg A)(a)$, the branch would be closed and a different conclusion could be reached. The assertion $(\neg B \sqcup \neg A)(a)$ will be added to the KB if either the subsumption relation $B \sqsubseteq \neg A$ or $A \sqsubseteq \neg B$ is asserted. This leads to the conclusion that when the assertions $A(a)$ and $B(a)$ are encountered, the agent should send the queries $A \sqsubseteq \neg B?$ and $B \sqsubseteq \neg A?$ to other agents and if it receives a positive or a default answer, the results of reasoning may change.

Example 1. Consider three agents with the following knowledge bases.

Agent 1	**Agent 2**	**Agent 3**
$\mathsf{Penguin}(\mathsf{PAT})$	$\mathsf{Penguin} \sqsubseteq \mathsf{Bird}$ $\dfrac{\mathsf{Bird} : \mathsf{Flies}}{\mathsf{Flies}}$	$\mathsf{Penguin} \sqsubseteq \neg\mathsf{Flies}$

The tableau reasoning procedure for the first agent will create the ABox $\mathcal{A} = \{\mathsf{Penguin}(\mathsf{PAT}), \neg\mathsf{Flies}(\mathsf{PAT})\}$. The first assertion is taken from the knowledge base and the second is the negation of the query. Now, the two queries that can be sent to other agents are $\mathsf{Penguin} \sqsubseteq \mathsf{Flies}$ and $\neg\mathsf{Flies} \sqsubseteq \neg\mathsf{Penguin}$. Any of these queries answered positively would directly cause the answer to the query $\mathsf{Flies}(\mathsf{PAT})$ to become positive. However, neither Agent 2 nor Agent 3 will answer positively. Agent 2 will nonetheless provide the answer $\dfrac{\mathsf{Penguin} : \mathsf{Flies}}{\mathsf{Flies}}$. The answer in the form of a default causes the asking agent to assimilate the default to its knowledge base. The agent must recompute the extensions, since adding a default can result in changing the number of extensions. Agent 1 can add this default to its knowledge base and compute a single extension $\{\mathsf{Penguin}(\mathsf{PAT}), \mathsf{Flies}(\mathsf{PAT})\}$, which can answer the query $\mathsf{Flies}(\mathsf{PAT})?$ positively.

The third agent's knowledge can, however, change this outcome since it knows that penguins do not fly. The default reasoning procedure, before deciding to apply the default (which came from Agent 2), will check its justification. Since this is done using the same tableau algorithm, the ABox $\mathcal{A} = \{\mathsf{Penguin}(\mathsf{PAT}), \mathsf{Flies}(\mathsf{PAT})\}$ will be tested for consistency. The first assertion of $\mathcal{A}$ comes from the knowledge base and the second one is the tested justification. This ABox will produce the queries $\mathsf{Penguin} \sqsubseteq \neg\mathsf{Flies}$ and $\mathsf{Flies} \sqsubseteq \neg\mathsf{Penguin}$. The first one of these can be answered by Agent 3 causing the tableau to close with a clash and disallowing the default $\frac{\mathsf{Penguin} : \mathsf{Flies}}{\mathsf{Flies}}$ to be applied for PAT. The final result is the answer FALSE to the query $\mathsf{Flies}(\mathsf{PAT})$.

The exchange of knowledge between agents is realized by extending the tableau reasoning procedure by adding a new tableau rule. The rule in Figure 2 says that if no other rules can be applied, pairs of concepts (A, B) are chosen to be formed into a query. The query is formed by creating two subsumptions $A \sqsubseteq \neg B$ and $B \sqsubseteq \neg A$. If a query succeeds, it results in making the tableau branch inconsistent.

If no other rules can be applied and $\mathcal{A}$ contains A and B, and the query (A, B) has not been issued,
then $\mathsf{prepareQuery}(A, B)$.

Fig. 2. Tableau rule for issuing queries

For each pair of assertions $A(a)$, $B(a)$, where A and B are atomic concepts and a is an individual, the procedure $\mathsf{prepareQuery}(A, B)$ is run. If either of the atomic concepts subsumes the other in the local knowledge base (i.e. $A \sqsubseteq B$ or $B \sqsubseteq A$), the queries are not sent, because a positive answer would lead to inconsistency. The subsumption test should be possible to be made very efficient by indexing the concept lattice [10], since only atomic concepts may appear in queries. The procedure $\mathsf{sendQuery}(a, q)$ asynchronously sends the query q to the agent a. Thus, the procedure prepareQuery returns immediately and does not wait to receive answers from queried agents.

The answers from other agents are recieved asynchronously. After all agents respond or a timeout is reached, if there are any answers, which are not FALSE, the local query has to be recomputed. In the process, new queries, which were not issued before can be sent. The process continues until no answers to queries are received. The process is guraranteed to finish, since no query is issued twice and the set of agents and their interface languages are finite.

The answerReceived function is a callback function called asynchronously after each answer from a remote agent is received. Note that the recomputation of extensions can be postponed until more answers are received in order to reduce the number of times the extensions are determined. When the agent is no longer waiting for answers from other agents, the answer to the query is determined by the local default reasoning procedure.

Algorithm 1. prepareQuery

Input: Concepts A, B

begin
  if $A \sqsubseteq B$ *or* $B \sqsubseteq A$ *is entailed by the local knowledge base* **then**
    return;
  targetAgents $= agents(A) \cap agents(B)$;
  foreach $a \in$ targetAgents **do**
    `sendQuery(`a, $A \sqsubseteq \neg B$`);`
    `sendQuery(`a, $B \sqsubseteq \neg A$`);`
end

Algorithm 2. answerReceived

Input: answer a

begin
  if a *is a default* $\frac{A : J \sqcap B}{B}$ **then**
    Add $\frac{A : J \sqcap B}{B}$ to the current knowledge base;
    Recompute extensions;
  else if a *is a positive answer* $A \sqsubseteq B$ **then**
    Add $\neg A \sqcup B$ to each $\mathcal{A}_i$ in the tableau;
end

6 Conclusion

The paper describes a model for DDL$\mathfrak{D}$-based multi-agent system. The distributed reasoning process is proposed. The formalism for default transformation is applied in order to achieve answers to subsumption queries that retain the information about the assumptions made during default reasoning. An algorithm based on Distributed Description Logic has been developed for reasoning in a multi-agent environment.

Default transformations can have an application to answering queries in a multi-agent system. Passing messages between agents in the form of defaults is more informative than strict answers, as the assumptions made during reasoning are not hidden from the querying agent, which in turn can itself validate the justifications to perform the inference locally.

References

1. Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.-C., Simon, L.: Distributed reasoning in a peer-to-peer setting: application to the semantic web. J. Artif. Int. Res. 25(1), 269–314 (2006)
2. Amir, E., McIlraith, S.: Partition-based logical reasoning for first-order and propositional theories. Artif. Intell. 162(1-2), 49–88 (2005)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York (2007)

4. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. Technical Report RR-93-20, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (1993)
5. Bao, J., Slutzki, G., Honavar, V.: A semantic importing approach to knowledge reuse from multiple ontologies. In: Proceedings of AAAI Conference 2007, pp. 1304–1309 (2007)
6. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. In: Spaccapietra, S., March, S., Aberer, K. (eds.) Journal on Data Semantics I. LNCS, vol. 2800, pp. 153–184. Springer, Heidelberg (2003)
7. Cuenca Grau, B., Parsia, B., Sirin, E.: Combining OWL ontologies using $\mathcal{E}$-connections. J. Web Semantics 4(1), 40–59 (2006)
8. Kaleta, M., Pałka, P., Toczyłowski, E., Traczyk, T.: Electronic trading on electricity markets within a multi-agent framework. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 788–799. Springer, Heidelberg (2009)
9. Kowalski, R., Sadri, F.: From logic programming towards multi-agent systems. Annals of Mathematics and Artificial Intelligence 25(3-4), 391–419 (1999)
10. Lewandowski, J., Rybinski, H.: A hybrid method of indexing multiple-inheritance hierarchies. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 211–220. Springer, Heidelberg (2009)
11. Reiter, R.: A logic for default reasoning. Artif. Intell. 13, 81–132 (1980)
12. Ryzko, D., Rybinski, H.: Distributed default logic for multi-agent system. In: IAT 2006: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Washington, DC, USA, pp. 204–210 (2006)
13. Ryżko, D., Rybinski, H., Więch, P.: Learning mechanism for distributed default logic based MAS - implementation considerations. In: Proceedings of the International IIS 2008 Conference, pp. 329–338 (2008)
14. Serafini, L., Tamilin, A.: Local tableaux for reasoning in distributed description logics. In: Proceedings of the 2004 International Workshop on Description Logics, DL 2004 (2004)
15. Więch, P., Rybiński, H.: A novel approach to default reasoning for MAS. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 484–493. Springer, Heidelberg (2010)
16. Więch, P., Rybiński, H.: Using default transformations for reasoning in MAS. Technical report, ICS, Warsaw University of Technology (2010)

Markov Blanket Approximation Based on Clustering

Paweł Betliński*

Institute of Informatics,
Warsaw University,
Banacha 2, 02-097 Warsaw, Poland
www.mimuw.edu.pl

Abstract. This paper presents new idea for Markov blanket approximation. It uses well known heuristic ordering of variables based on mutual information, but in another way then it was considered in previous works. Instead of using it as a simple help tool in a more complicated method most often based on statistical tests - presented here idea tries to rely without any further statistical tests only on the heuristic and its previously not considered interesting properties.

Keywords: Markov blanket, Bayesian network, mutual information, clustering.

1 Introduction

Markov blanket is one of the essential concepts in the domain of Bayesian networks (see [4], [3])). It has first time appeared in 1988 in Judea Pearl's work ([4]). Let $\underline{A}$ be the vector of discrete random variables. In practice we consider $\underline{A}$ as a set of attributes of some information system. Markov blanket of some variable $X \in \underline{A}$, denoted as $MB(X)$, is a subset $MB(X) \subseteq \underline{A} \setminus \{X\}$ such that $X \perp (\underline{A} \setminus (\{X\} \cup MB(X))) \mid MB(X)$ (where $U \perp V \mid W$ means conditional independence of U and V given W). What we are usually most interested about are minimal Markov blankets with respect to the relation $\subset$. There is a special name for such minimal Markov blankets - they are called Markov boundaries, but it is the term rather not often used. Many authors instead of Markov boundary simply call this minimal forms as Markov blankets. I will also use this convention.

* Research task SYNAT: ,,Establishment of the universal, open, hosting and communication, repository platform for network resources of knowledge to be used by science, education and open knowledge society". Strategic scientific research and experimental development program: ,,Interdisciplinary System for Interactive Scientific and Scientific-Technical Information", Grant No. SP/I/77065/10. This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: ,,Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 192–202, 2011.

It turns out, that for big family of possible distributions over $\underline{A}$ Markov blanket is unique. First of all, it is unique for every discrete distribution for which exists so called perfect map (I only mention about it in this article, definition can be found in [3], from page 95). This property is closely related to Bayesian networks domain, but there is also other property, which can be understood without any further knowledge about this topic. Suppose that probability distribution of discrete random variables in $\underline{A}$ is strictly positive . That is, every value configuration of variables in $\underline{A}$ has probability greater than zero (for each variable we consider all its values within its state space). Then for every $X \in \underline{A}$ there is a unique Markov blanket of X (proof of this result can be found in [4]).

This paper will not consider Markov blankets applications, but it is worth to sketch two of them. One is attribute selection: if we want to predict decision variable given the rest of attributes we can simply limit this task to predict decision on the basis of its Markov blanket - in probabilistic sense we don't loose any information. This application attracted more attention after Koller and Sahami's work in 1996 ([1]). The second application is associated with the domain from which Markov blankets come - Bayesian networks. It turns out that knowledge of Markov blanket of each variable of some random vector (in practice - set of attributes of information system) is in theory sufficient to induce Bayesian network for this variables. Moreover, assuming bounded size of Markov blankets, there exist time efficient algorithms for inducing Bayesian network from Markov blankets (see [2]).

In 1999 in the same paper ([2]) there also appears important algorithm for Markov blanket approximation - Grow-Shrink. It is based on statistical tests for conditional independence, and has been theoretically proved to return exact solution assuming that statistical tests don't make mistakes. Of course these tests makes mistakes - so in real application this method returns only some approximation of Markov blanket. Later in 2003 there has appeared IAMB algorithm (Incremental Association Markov Blanket , [5]) which is actually quite simple modification of Grow-Shrink, but resulting in significantly better accuracy. The modification was most of all to replace one of the element of the method: simple and time efficient heuristic ordering of variables according to its influence on the variable for which we want to find Markov blanket (main cost is here just ordering sequence of numbers), with time less efficient (pessimistic quadratic in number of variables) but much better ordering based on more advanced heuristic. The crucial reason why IAMB is better is this advanced heuristic ordering, which significantly reduces number of statistical tests for conditional independence necessary to conduct, and reduces number of conditional variables in these tests (which results in greater reliability of these tests). Both methods Grow-Shrink and IAMB were beginning of constraint-based type stream of methods for Markov blanket approximation - which is characterized by using conditional independence tests. These type of methods were definitely most popular in last decade. The inspiration for them was to reduce as much as possible number of necessary tests and number of conditional variables in these tests - as it was a bit done in IAMB, because this could lead to better accuracy.

The method which is proposed in this article lies outside of this constraint-based stream, but gets inspiration from the method which is inside - from IAMB. The same heuristic ordering of variables as in IAMB is here used, but the purpose to do this is not the same as in IAMB. Here no further tests will be performed, but instead of this heuristic ordering is here the main tool on which the method relies. It turns out that there are a few interesting properties of this IAMB variables ordering, which were not considered and used in any method. The approach in this paper tries to take advantage from them.

This paper is organized as follows: section 2 describes in particular IAMB heuristic ordering and its properties on which rely described in section 3 new methods. Section 4 presents experiments comparing proposed algorithms with some already well known. This section also summarize whole approach.

2 Theoretical Aspects

The aim of this section is to present some theoretical facts which are the basis for proposed in section 3 methods of Markov blanket approximation.

The IAMB variables ordering is built on some arbitrary chosen measure of dependence between two variables - IAMB authors used in their experiments popular statistical measure - mutual inforamation.

Definition 1. *Let X and Y be discrete random variables with state spaces respectively $\{x_1, ..., x_s\}$ and $\{y_1, ..., y_t\}$. Mutual information of X and Y, which we will denote as $M(X, Y)$, is equal to*

$$M(X, Y) = H(Y) - \mathbb{E}H(Y \mid X),$$

where $H(Y) = -\sum_{i=1}^{t} Pr(Y = y_i) \log(Pr(Y = y_i))$ is an entropy measure, $H(Y \mid X = x_j) = -\sum_{i=1}^{t} Pr(Y = y_i \mid X = x_j) \log(Pr(Y = y_i \mid X = x_j))$, and $\mathbb{E}H(Y \mid X) = \sum_{j=1}^{s} H(Y \mid X = x_j) Pr(X = x_j)$

Some well known facts about this measure are that $M(X, Y) = M(Y, X)$, $M(X, Y) \geq 0$, and $M(X, Y) = 0 \iff X \perp Y$. Intuitively, the closer to zero is mutual information measure, the closer to independence are considered variables.

In Grow-Shrink algorithm variables are ordered simply in descending order by the value of their mutual information with variable for which we want to find Markov blanket - let T be this variable. In IAMB the ordering takes care about common influences of sets of variables on T.

Assume that we have discrete random vector $\{X_1, X_2, ..., X_n, T\}$, where T is the variable for which we want to find $MB(T)$ - its Markov blanket. The values on which IAMB ordering relies we define as $M_i = M(T, \{X_1, X_2, ..., X_i\})$, for $i \in \{1, ..., n\}$ (where $\{X_1, X_2, ..., X_i\}$ we understand here as one random variable in vector form). Additionally we define $M_0 = 0$. Then:

Fact 1. *Let's denote* $C_i = \{X_1, X_2, ..., X_i\}$ *for* $i \in \{1, ..., n\}$, $C_0 = \emptyset$. *Then:*

(a) $M_0 \leq M_1 \leq M_2 \leq ... \leq M_n$.
(b) For $1 \leq i \leq n$ *we have that* $M_i = M_{i-1} \iff X_i \perp T \mid C_{i-1}$.
(c) Assume additionally that $MB(T)$ *is unique. If* $j = \max\{i \in \{1, ..., n\} : M_i > M_{i-1}\}$, *then* $X_j \in MB(T)$ *and* $MB(T) \subseteq \{X_1, X_2, ..., X_j\}$. *If such* j *doesn't exist, that is* $M_1 = M_2 = ... = M_n = 0$, *then* $MB(T) = \emptyset$.

Proof. (c) Assume that for some $0 \leq i < j$ we have that $MB(T) \subseteq C_i$. This would of course mean that $\{X_{i+1}, X_{i+2}, ..., X_n\} \perp T \mid C_i$, so in particular $X_j \perp T \mid C_{j-1}$ - but this is not possible, because of *(b)*.

So by contradiction we have proved, that for every $0 \leq i < j$ $MB(T) \not\subseteq C_i$. So to prove *(c)* it is of course sufficient to show that $MB(T) \subseteq C_j$.

If $j = n$, then clearly $MB(T) \subseteq C_j$. So let's assume that $j < n$. We have that $M_j = M_n$. Let's consider instead of sequence of variables $X_1, X_2, ..., X_n$ the sequence $X_1, X_2, ..., X_j, \{X_{j+1}, X_{j+2}, ..., X_n\}$, where we treat $\{X_{j+1}, X_{j+2}, ..., X_n\}$ as one variable in vector form. Corresponding mutual information values to this sequence are of course $M_1, M_2, ..., M_j, M_n$. Now we simply use *(b)* and obtain, that $\{X_{j+1}, X_{j+2}, ..., X_n\} \perp T \mid \{X_1, X_2, ..., X_j\}$ - this is one of two Markov blanket properties (second property is minimality). From assumption we have that $MB(T)$ is unique, which means that for sure $MB(T) \subseteq \{X_1, X_2, ..., X_j\}$.

In the case when j doesn't exist, we have in particular that $M_n = 0$, which means that $\{X_1, X_2, ..., X_n\} \perp T$, so $MB(T) = \emptyset$. □

Part *(a)* and *(b)* are well known, so I omit their proof (which is actually simple consequence of previously written mutual information properties).

Now let's assume, that variables $X_1, ..., X_n$ are already ordered according to IAMB heuristic. This means, that for $i \in \{1, ..., n\}$:

$$X_i = \underset{X \in \{X_i, X_{i+1}, ..., X_n\}}{\operatorname{argmax}} M(T, \{X_1, X_2, ..., X_{i-1}, X\})$$

(for $i = 1$ we understand $M(T, \{X_1, X_2, ..., X_{i-1}, X\})$ as $M(T, X)$) With this assumption it turns out, that we can reinforce part *(c)* in previous fact:

Fact 2. *Assume that* $MB(T)$ *is unique, and that variables* $X_1, ..., X_n$ *are already ordered according to IAMB heuristic. Let* $M_i = M(T, \{X_1, X_2, ..., X_i\})$, *for* $i \in \{1, ..., n\}$, *and* $j = \max\{i \in \{1, ..., n\} : M_i > M_{i-1}\}$. *If such* j *exists and* $j > 1$ *then* $X_{j-1} \in MB(T)$.

Proof. Let's consider modified sequence $X'_1 = X_1, X'_2 = X_2, ..., X'_{j-2} = X_{j-2}$, $X'_{j-1} = X_j, X'_j = X_{j-1}, X'_{j+1} = X_{j+1}, X'_{j+2} = X_{j+2}, ..., X'_n = X_n$ - so we swap X_{j-1} and X_j. Corresponding mutual information values to this sequence are $M'_1 = M_1, M'_2 = M_2, ..., M'_{j-2} = M_{j-2}, M'_{j-1}, M'_j = M_j, M'_{j+1} = M_{j+1}, ..., M'_n = M_n$ - clearly the only change might be on $j - 1$ place. From assumption order $X_1, ..., X_n$ is according to IAMB heuristic, so $M'_{j-1} \leq M_{j-1}$. Also from assumption we have that $M_j > M_{j-1}$. This gives us $M'_{j-1} < M_j$, so $M'_{j-1} < M'_j$.

That is we have that $\max\{i \in \{1, ..., n\} : M'_i > M'_{i-1}\} = j$, so from fact 1 *(c)* we have that $X'_j \in MB(T)$, but $X'_j = X_{j-1}$. □

Fact 1 *(c)* and fact 2 were not considered and used in any previous Markov blanket approximation approaches. In the next section there are proposed methods which tries to do this.

3 Proposed Methods

We can treat IAMB ordering of variables as a greedy approach which tries to place $MB(T)$ in at most number of first positions of this order as possible. The ideal situation would be of course if for $|MB(T)| = k$ on the first k positions of this order are exactly all variables from $MB(T)$ - IAMB heuristic doesn't guarantee this (it is easy to show examples of distributions of variables $\{X_1, X_2, ..., X_n, T\}$ for which IAMB order is not perfect), although it is still really good, especially comparing to some previous heuristic orderings - like in Grow-Shrink.

But obtaining this order is only the first step - the second would be to somehow obtain from it approximation of $MB(T)$, let's denote it as $\widehat{MB(T)}$. In IAMB the main tools to catch $\widehat{MB(T)}$ from this order are statistical conditional independence tests.

What here will be proposed is to obtain $\widehat{MB(T)}$ from IAMB order without any tests, instead of them looking not only at the order, but also at the mutual information values corresponding to it: $M_1, M_2, ..., M_n$.

The basic idea to do this is to directly use fact 1 *(c)*. In theory it would be very simple: Let's assume that we have variables $\{X_1, X_2, ..., X_n, T\}$, where for variable T we want to find its Markov blanket approximation $\widehat{MB(T)}$. We assume also that variables $X_1, X_2, ..., X_n$ are already ordered according to IAMB heuristic. As previously we denote $M_i = M(T, \{X_1, X_2, ..., X_i\})$ for $i \in \{1, ..., n\}$, $M_0 = 0$. Then we find $j = \max\{i \in \{1, ..., n\} : M_i > M_{i-1}\}$ and return $\widehat{MB(T)} = \{X_1, X_2, ..., X_j\}$, or if such j doesn't exist - we return $\widehat{MB(T)} = \emptyset$. In theory such approach guarantees us that we catch as $\widehat{MB(T)}$ the smallest number of first variables in IAMB order which contain $MB(T)$.

All this procedure seems to be very simple, but in fact it is simple only in theory. The main problem which we have in practice is that we don't have exact information about distribution of $\{X_1, X_2, ..., X_n, T\}$, but only some approximate information in the form of finite table. This means that IAMB order is obtained only approximately, and also values $M_1, ..., M_n$ are approximate. Moreover, value of M_i looses rapidly its reliability with increasing i (When i is increasing, variables $\{X_1, X_2, ..., X_i\}$ which appear in $M_i = M(T, \{X_1, X_2, ..., X_i\})$ divide finite number of examples - rows in table into fast increasing number of parts according to all possible value configurations of them, so sizes of this parts decrease fast and make calculation of entropy of T on each of them less reliable.).

Now the question arises, whether, despite those problems, it is possible to obtain somehow in practice good approximation of $j = \max\{i \in \{1, ..., n\} : M_i > M_{i-1}\}$? The best way to answer this question is to show the following examples.

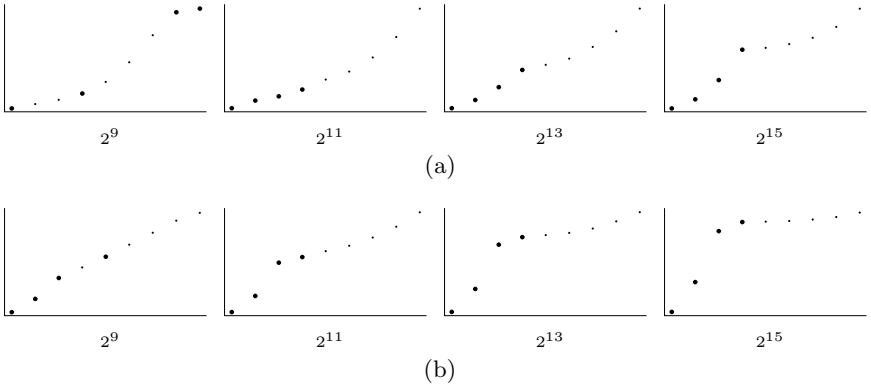


Fig. 1.

Both examples 1 (Fig. 1 (a)) and 2 (Fig. 1 (b)) corresponds to two another randomly chosen distributions over binary variables $\{X_1, X_2, ..., X_9, T\}$. For both distributions we know $MB(T)$ - it corresponds to big points on the pictures. For each of this two distributions there were artificially generated from them four tables of size (number of rows) 2^9, 2^{11}, 2^{13}, 2^{15} (how to obtain all these things is described in next section). Each picture presents IAMB order of variables $X_1, X_2, ..., X_9$ (order is from left to right) calculated on the basis of the table of size corresponding to the caption. We can't read all this order - we only see this what we are interested: positions of variables in $MB(T)$ - big points. Assuming that on some picture the order of variables is $X'_1, X'_2, ..., X'_9$ (some permutation of sequence $X_1, X_2, ..., X_9$), the height of i-th from the left point on that picture corresponds to $M(T, \{X'_1, X'_2, ..., X'_i\})$.

Although the pictures are not exact - they are only approximation of real values, of course this approximation becomes better with increase of size of table - it turns out that for this greater table sizes human can easily recognize where is the crucial position corresponding in theory to $j = \max\{i \in \{1, ..., n\} : M_i > M_{i-1}\}$. Sometimes it is easier to recognize it - as in example 1, where we can already see this position for table of size 2^{13}. Sometimes it is harder - like in example 2. Here it is easier to make mistake and say that rather $j = 3$, while it should be $j = 4$.

Here we reach to the main question - how to construct algorithm which will recognize with big accuracy the correct position j ? Is it easier to characterize this position with respect to the left side of j on the picture (the most reliable values), or to the right side of j (the least reliable values)?

Answer might be a bit surprising - it seems that it is much easier to recognize it with respect to the right side. The reason is following: Left side, although the most reliable, has in general irregular shape - as its theoretical ideal form is. Right side, although the least reliable, is in theory a straight horizontal line, which in a finite table consideration looks most often as a line quite smoothly

with decrease of reliability curving up - so the position j should be the first one going from right to left after which this behavior is changed.

So proposed further method recognize position j from the right side of the picture, which somehow breaks the principle of many previous approaches: to rely on the most reliable measures and tests.

Method of recognizing desired inflection position j:

- Assume that $MB(T)$ is unique, and that variables $X_1, X_2, ..., X_n$ are already ordered according to IAMB heuristic.
- We use normalized version of the graph, where the height of i-th point $M_i = M(T, \{X_1, ..., X_i\})$ is additionally divided by $H(T)$, we will denote this normalized height as N_i, and define $N_0 = 0$. We name all this points simply: $X_1, X_2, ..., X_n$, the coordinates of point X_i on the graph is (i, N_i). Additionally we define point $X_0 = (0, N_0) = (0, 0)$.
- We define also for $i \in \{1, ..., n-1\}$ $v_i = \arctan(N_i - N_{i-1}) - \arctan(N_{i+1} - N_i)$. If we understand angle $\sphericalangle X_{i-1}X_iX_{i+1}$ as the lower one, then it is easy to see that $\sphericalangle X_{i-1}X_iX_{i+1} = \pi - v_i$.
- The aim: Find the first point going from right to left such, that it is significant inflection point into the concave shape comparing to most often convex flexions of the points on the right.
- The procedure: Starting from N_START-th point counting from the right side, moving in each next iteration by one point to the left, we procede the following: Assume, that we are in point X_k. If $v_k > 0$ (so the flexion is into the concave shape), then we perform some clusterization of values $v_k, v_{k+1}, ..., v_{n-1}$ into two groups. If v_k turns out to be one cluster, then we stop algorithm and return position k as a desired point of inflection.
- If the iteration goes through points from right to left including X_1 and doesn't find any point of inflection - we return "no position of inflection".

N_START is arbitrary chosen parameter. It is necessary, because we need to compare somehow considered point with the previous on the right to detect whether it is point of inflection. Another parameter - very important - is method of clustering. In the next section I describe what method I used in experiments.

One of two proposed methods of approximation $MB(T)$ we have already described: If sketched above algorithm returns as a point of inflection some position l - we return $\widehat{MB(T)} = \{X_1, X_2, ..., X_l\}$, or if algorithm returns "no point of inflection" - we return $\widehat{MB(T)} = \emptyset$. The second proposed method is more complex and can be proved to return exact solution $MB(T)$ if all measures are calculated not approximately on the basis of some finite table, but we have their exact values. We assume that $MB(T)$ is unique. The method is following:

1. Let $X'_1, X'_2, ..., X'_n$ be IAMB order of variables $X_1, X_2, ..., X_n$.
2. Use the method of recognizing desired inflection position. If the method returns "no point of inflection", then stop algorithm and return $\widehat{MB(T)} = \emptyset$. If the method returns position $l = 1$, then stop algorithm and return

$\widehat{MB(T)} = X_1$. If $l = 2$ - stop and return $\widehat{MB(T)} = \{X_1, X_2\}$. If $l > 2$, then: assign $X_1 = X'_1$, $X_2 = X'_2$, ..., $X_{l-2} = X'_{l-2}$, $X_{l-1} = \{X'_{l-1}, X'_l\}$, $X_l = X'_{l+1}$, $X_{l+1} = X'_{l+2}$, ..., $X_{n-1} = X'_n$, assign $n = n - 1$ and go to step 1.

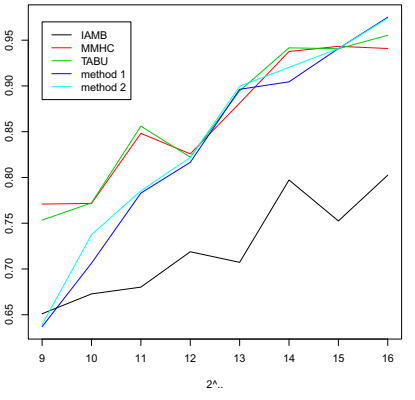
The method is somehow similar to agglomerative clustering. In every step we 'glue' two variables X'_{l-1} and X'_l treating them as one in a vector form. In theoretical consideration (all measures exact) from fact 2 we know that both variables X'_{l-1} and X'_l belongs to $MB(T)$ - that's why we can 'glue' them. In the next step we have one variable less - but Markov blanket remaines the same - the only (virtual) change is that instead of X'_{l-1} and X'_l it contains one variable $\{X'_{l-1}, X'_l\}$. So in theory after whole this procedure all variables in $MB(T)$ and only them are connected into one variable in vector form - which is returned. Of course in practice all used measures are not calculated exactly but approximately - so whole method is also some approximation of $MB(T)$.

4 Experiments and Summary

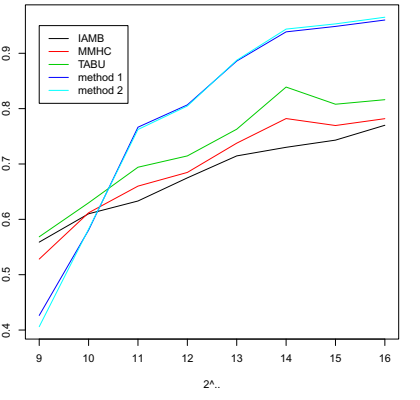
Each presented further experiment has the following schema: Parameters are: s - increasing finite sequence of natural numbers; n, RAND_REP, CLUST_REP, MAX_MB $\in \mathbb{N}$; $d \in [0, 1]$. For each element $s[i]$ of s repeat RAND_REP times:

- Randomly choose directed acyclic graph $\mathbb{G}$ of n vertices and density d, that is each pair of vertices is connected with probability d. Assign to each node X of $\mathbb{G}$ - which will represent binary variable - randomly (uniformly) chosen parameters describing conditional probability of X given its parents in $\mathbb{G}$. $\mathbb{G}$ together with parameters is our random Bayesian network $\mathbb{BN}$ of density d.
- After such randomly chosen parameters for $\mathbb{G}$ with probability 1 the distribution holded by $\mathbb{BN}$ - let's call it $\mathbb{P}_{\mathbb{BN}}$ - has perfect map in the form of exactly graph $\mathbb{G}$ (see [3]). According to another well known fact (see [3]) this means, that for every node X of $\mathbb{G}$ we can read $MB(X)$ for distribution $\mathbb{P}_{\mathbb{BN}}$ directly from graph $\mathbb{G}$ ($MB(X)$ is a set of parents, children and parents of children of X). So for our randomly chosen distribution holded by network $\mathbb{BN}$ we know in advance what is the real Markov blanket of each variable.
- We choose random node T of $\mathbb{G}$. We generate table of $s[i]$ rows - each row independly generated from distribution $\mathbb{P}_{\mathbb{BN}}$ holded by network $\mathbb{BN}$ (using standard procedure included in logic sampling (see [3], page 215)).
- As accuracy of each Markov blanket approximation method on this table we return $\frac{|\widehat{MB(T)} \cap MB(T)|}{|\widehat{MB(T)} \cup MB(T)|}$ if $|\widehat{MB(T)} \cup MB(T)| > 0$, otherwise we return 1.

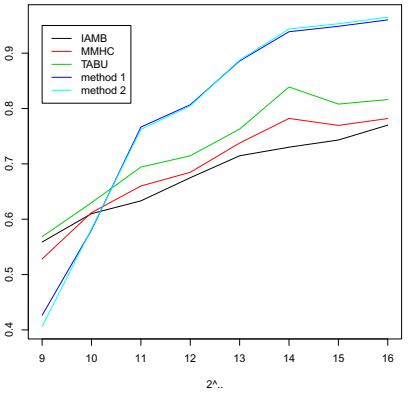
As accuracy of given method on the tables of size $s[i]$ we return average accuracy of the method in RAND_REP described repetitions. Parameter MAX_MB is the maximal considered size of Markov blankets. That is, if in some repetition it turns out that $|MB(T)| >$ MAX_MB - the repetition is cancelled and repeated until $|MB(T)| \leq$ MAX_MB. Both proposed in previous section methods use described algorithm to recognize point of inflection. The important



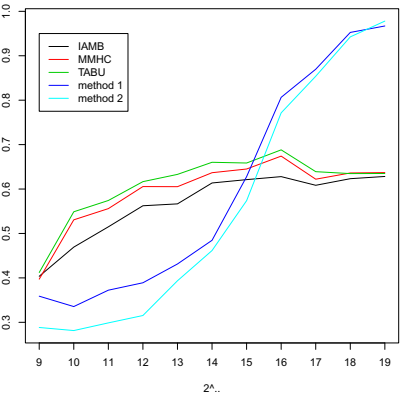
d=0.1



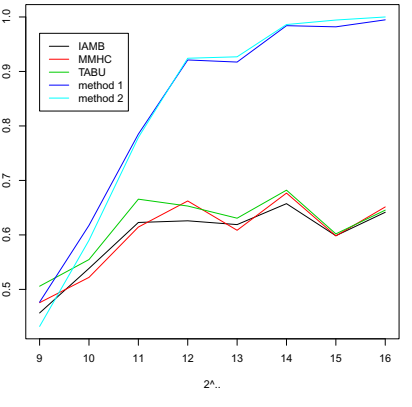
n=10



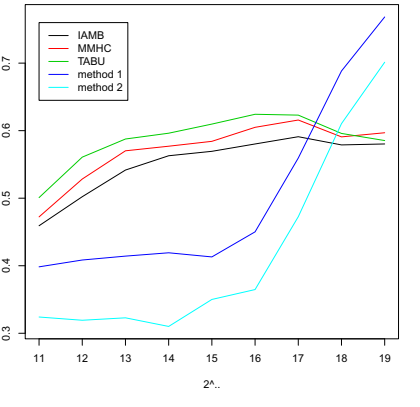
d=0.3



n=15



d=0.5



n=20

(a) (b)

Fig. 2. (a) n=10, N_START=4, RAND_REP=200, CLUST_REP=100, MAX_MB=6 (b) N_START=4, RAND_REP=200, CLUST_REP=100, MAX_MB=n-4, d=0.3

element of this algorithm is clustering method - I have used 2-means clustering repeated with random start configuration CLUST_REP times. Then simply most often result is chosen. N_START is parameter described in previous section.

Fig. 2 (a) and Fig. 2 (b) presents results of two experiments based on described schema. In the first one we consider situation when for the constant number of variables ($n = 10$), the density of dependencies is increased ($d = 0.1, 0.3, 0.5$, this affect increase of size of Markov blankets). In the second one density is constant ($d = 0.3$), and we increase number of variables ($n = 10, 15, 20$). X-axis on each picture is sequence s, Y-axis is accuracy of the methods. Method 1 is first proposed simple method, while method 2 is the second agglomerative method. MMHC (Max-Min Hill-Climbing, [6]) is a popular, strong hybrid (score and constraint-based) method of approximation Bayesian network - but from the obtained network we can easy read approximation of Markov blanket of any variable. TABU is a score-based Bayesian network learning approach implemented in package *bnlearn* ([7]) for statistical tool R ([8])).

First of all we can see that methods 1 and 2 behave similarly, but I prefer method 1, because of its simplicity: my implementation of this method has time complexity $O(n^2 m \log m)$ where m is number of rows and n is number of columns of the table, while method 2 is $O(n^3 m \log m)$.

Secondly, we can see from results in Fig. 2 (a) that both methods become stronger then other when density of dependencies is increasing. Especially interesting is second picture ($d = 0.3$). It shows that for small statistical significance of the tables better are well known approaches, but when it is big - proposed methods can achieve much better accuracy then other. For IAMB and MMHC I could see the reason in their weak point - statistical tests. Method 1 is similar to IAMB but instead of tests it uses simple picture recognition procedure - which seems to overtake tests accuracy when statistical significance of the tables is increasing. In second experiment (Fig. 2 (b)) with constant density 0.3 we see similar phenomenon for increasing number of variables.

Summarizing, it is possible with proposed approach to obtain significantly better accuracy, especially when density of dependencies is big, but it is possible only with sufficiently big statistical significance of the tables - this condition unfortunately in practice become hard with increasing number of variables. However, the idea should be useful in many specific applications where we have data which with high quality illustrates distribution of variables.

References

1. Koller, D., Sahami, M.: Toward Optimal Feature Selection, Technical Report. Stanford InfoLab (1996)
2. Margaritis, D., Thrun, S.: Bayesian Network Induction via Local Neighborhoods. In: Proceedings of the Neural Information Processing Systems, pp. 505–511 (1999)
3. Neapolitan, R.: Learning Bayesian Networks. Prentice Hall, Upper Saddle River (2004)

4. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
5. Tsamardinos, I., Aliferis, C., Statnikov, A.: Algorithms for Large Scale Markov Blanket Discovery. In: Proceedings of the FLAIRS (2003)
6. Tsamardinos, I., Brown, L., Aliferis, C.: The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. Machine Learning 65(1), 31–78 (2006)
7. http://www.bnlearn.com
8. http://www.r-project.org

Tri-Based Set Operations and Selective Computation of Prime Implicates*

Andrew Matusiewicz, Neil V. Murray, and Erik Rosenthal

Institute for Informatics, Logics & Security Studies, Department of Computer Science,
University at Albany – SUNY, Albany, NY 12222, USA
{a_matusi,nvm}@cs.albany.edu,
erik.rosenthal@jeanerik.net

Abstract. A more efficient version of the prime implicate algorithm introduced in [12] that reduces the use of subsumption is presented. The algorithm is shown to be easily tuned to produce restricted sets of prime implicates. Experiments that illustrate these improvements are described.

1 Introduction

Prime implicants were introduced by Quine [15] as a means of simplifying propositional logical formulas in disjunctive normal form (DNF); prime implicates play the dual role in conjunctive normal form (CNF). Implicants and implicates have many applications, including abduction and program synthesis of safety-critical software [8]. All prime implicate algorithms of which the authors are aware make extensive use of clause set subsumption; improvements in both the pi-trie algorithm and its core subsumption operations are therefore relevant to all such applications.

Numerous algorithms have been developed to compute prime implicates — see, for example, [1, 2, 4–7, 9, 10, 14, 16, 20, 21]. Most use clause sets or truth tables as input, but rather few allow arbitrary formulas, such as the pi-trie algorithm introduced in [12]. Some are incremental in that they compute the prime implicates of a clause conjoined to a set of prime implicates. This recursive algorithm stores the prime implicates in a trie — i.e., a labeled tree — and has a number of interesting properties, including the property that, at every stage of the recursion, once the subtrie rooted at a node is built, some superset of each branch in the subtrie is a prime implicate of the original formula. This property admits variations of the algorithm that compute restricted sets of prime implicates, such as only positive ones or those not containing specific variables. These variations significantly prune the search space, and experiments indicate that significant speedups are obtained. In this paper, the improvements and properties are introduced and explained; experimental results are provided.

Basic terminology and the fundamentals of pi-tries are summarized in Section 2. The analysis that leads to the new pi-trie algorithm is developed in Section 3, and trie-based

* This research was supported in part by the National Science Foundation under grants IIS-0712849 and IIS-0712752, and by the Air Force Research Laboratory, Award No. FA8750-10-1-0206, Subaward No. 450044-19191.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 203–213, 2011.

set operations and experiments with them are described in Section 4. The new pi-trie algorithm and the results of experiments that compare the new algorithm with the original are described in Section 5. Useful properties of the algorithm are established in Section 6, and results of experiments that illustrate efficiency improvements are presented in Section 7. Proofs of lemmas and theorems can be found in [13].

2 Preliminaries

The terminology used in this paper for logical formulas is standard: An *atom* is a propositional variable, a *literal* is an atom or the negation of an atom, and a *clause* is a disjunction of literals. Clauses are often referred to as sets of literals; however, it is sometimes convenient to represent them with logical notation. An *implicate* of a formula is a clause entailed by the formula, and a non-tautological clause is a *prime implicate* if no proper subset is an implicate. The set of prime implicates of a formula $\mathcal{F}$ is denoted $\mathcal{P}(\mathcal{F})$. Note that a tautology has no prime implicates, and the empty clause is the only prime implicate of a contradiction.

The trie is a tree in which each branch represents the sequence of symbols labeling the nodes (or sometimes edges) on that branch, in descending order. The nodes along each branch represent the literals of a clause, and the conjunction of all such clauses is a CNF equivalent of the formula. If there is no possibility of confusion, the term *branch* will often be used for the clause it represents. Further, it will be assumed that a variable ordering has been selected, and that nodes along each branch are labeled consistently with that ordering. A trie that stores all prime implicates of a formula is called a *prime implicate trie*, or simply a pi-trie. It is convenient to employ a ternary representation of pi-tries, with the root labeled 0 and the ith variable appearing only at the ith level. If $v_1, v_2, \ldots, v_n$ are the variables, then the children of a node at level i are labeled v_{i+1}, $\neg v_{i+1}$, and 0, left to right. With this convention, any subtrie (including the entire trie) is easily expressed as a four-tuple consisting of its root and the three subtries. For example, the trie $\mathcal{T}$ can be written $\langle r, \mathcal{T}^+, \mathcal{T}^-, \mathcal{T}^0 \rangle$, where r is the label of the root of $\mathcal{T}$, and $\mathcal{T}^+$, $\mathcal{T}^-$, and $\mathcal{T}^0$ are the three (possibly empty) subtries. The ternary representation will generally be assumed here. Lemma 1 is well known and stated without proof.

Lemma 1. Clause set $\mathcal{S}$ is prime iff $\mathcal{S}$ is a resolution-subsumption fixed point. □

3 Prime Implicates under Truth-Functional Substitution

The pi-trie algorithm computes $\mathcal{P}(\mathcal{F})$ recursively from $\mathcal{P}(\mathcal{F}[\alpha/v])$, where α is a truth constant 0 or 1 and is substituted for every occurrence of the variable v in $\mathcal{P}(\mathcal{F})$. When no confusion is possible, $\mathcal{F}[0/v]$ and $\mathcal{F}[1/v]$ will be denoted $\mathcal{F}_0$ and $\mathcal{F}_1$, respectively.

To transform $\mathcal{P}(\mathcal{F}_0)$ and $\mathcal{P}(\mathcal{F}_1)$ into $\mathcal{P}(\mathcal{F})$, note first that $\mathcal{F} \equiv (v \vee \mathcal{F}_0) \wedge (\neg v \vee \mathcal{F}_1)$; i.e., $\mathcal{P}(\mathcal{F})$ is logically equivalent to $(v \vee \mathcal{P}(\mathcal{F}_0)) \wedge (\neg v \vee \mathcal{P}(\mathcal{F}_1))$. Let J_0 and J_1, respectively, be the clause sets produced by distributing v — respectively, $\neg v$ — over $\mathcal{P}(\mathcal{F}_0)$ — respectively, $(\mathcal{P}(\mathcal{F}_1))$. Observe that J_0 and J_1 are (separately) resolution-subsumption fixed points because $\mathcal{P}(\mathcal{F}_0)$ and $\mathcal{P}(\mathcal{F}_1)$ are. Subsumption cannot hold between one clause in J_0 and one in J_1 since one contains v and the other $\neg v$. Thus if

$J_0 \cup J_1$ is not a resolution-subsumption fixed point, producing a fixed point from it must require resolutions having one parent from each. These can be restricted to resolving on v and $\neg v$ because any other produces a tautology. Note that each such resolvent is the union of a clause from $\mathcal{P}(\mathcal{F}_0)$ and one from $\mathcal{P}(\mathcal{F}_1)$. This proves

Lemma 2. The only useful resolvents in $J_0 \cup J_1$ are on v and $\neg v$. □

It turns out to be sufficient to consider all such resolvents but no others:

Theorem 1. Let $\mathcal{F}$ be a logical formula and let v be a variable in $\mathcal{F}$. Suppose E is a prime implicate of $\mathcal{F}$ not containing v. Then $E = (C \cup D)$, where $C \in \mathcal{P}(\mathcal{F}_0)$ and $D \in \mathcal{P}(\mathcal{F}_1)$. □

Theorem 1 and the discussion leading up to it suggest a method for computing $\mathcal{P}(\mathcal{F})$ from $\mathcal{P}(\mathcal{F}_0)$ and $\mathcal{P}(\mathcal{F}_1)$. From Lemma 2, no useful resolvent can contain v or $\neg v$. Thus the prime implicates that do are already in $J_0 \cup J_1$. By Theorem 1 the useful resolvents account for all prime implicates of $\mathcal{F}$ that do not contain the variable v. Thus, to produce $\mathcal{P}(\mathcal{F})$, it suffices to obtain the subsumption minimal subset of the fully resolved $J_0 \cup J_1$. Denote $\mathcal{P}(\mathcal{F}_0)$ and $\mathcal{P}(\mathcal{F}_1)$ by $\mathcal{P}_0$ and $\mathcal{P}_1$, respectively, and partition each into two subsets: $\mathcal{P}_0^{\supseteq}$, the clauses in $\mathcal{P}_0$ subsumed by some clause in $\mathcal{P}_1$, and $\mathcal{P}_0^{\bowtie}$, the remaining clauses in $\mathcal{P}_0$. Define $\mathcal{P}_1^{\supseteq}$ and $\mathcal{P}_1^{\bowtie}$ similarly.

Theorem 2. Let J_0, J_1, $\mathcal{P}_0$, $\mathcal{P}_1$, $\mathcal{P}_0^{\supseteq}$, $\mathcal{P}_0^{\bowtie}$, $\mathcal{P}_1^{\supseteq}$, and $\mathcal{P}_1^{\bowtie}$ be defined as above. Then

$$\mathcal{P}(\mathcal{F}) = (v \vee \mathcal{P}_0^{\bowtie}) \cup (\neg v \vee \mathcal{P}_1^{\bowtie}) \cup \mathcal{P}_0^{\supseteq} \cup \mathcal{P}_1^{\supseteq} \cup \mathcal{U} \qquad (*)$$

where $\mathcal{U}$ is the maximal subsumption-free subset of $\{C \cup D \mid C \in \mathcal{P}_0^{\bowtie}, D \in \mathcal{P}_1^{\bowtie}\}$ in which no clause is subsumed by a clause in $\mathcal{P}_0^{\supseteq}$ or in $\mathcal{P}_1^{\supseteq}$. □

Observe that if C in $\mathcal{P}_0$ strictly subsumes D in $\mathcal{P}_1^{\supseteq}$, then C is in $\mathcal{P}_0^{\bowtie}$: Were C in $\mathcal{P}_0^{\supseteq}$, the clause in $\mathcal{P}_1$ that subsumes C would also subsume D, which is impossible. As a result, C appears in the prime implicate $\{v\} \cup C$, which does not subsume D.

4 Tri-Based Operations

The pi-trie algorithm introduced in [12] used a routine called PIT, which was developed with a branch by branch analysis. The new version of the algorithm uses a new version of PIT. The two appear different but in fact use similar methods to construct $\mathcal{P}(\mathcal{F})$ from $\mathcal{P}_0$ and $\mathcal{P}_1$. The new one employs the set operations of Theorem 2. The resulting development is arguably more intuitive, and the new algorithm is more efficient.

One improvement comes from identifying $\mathcal{P}_0^{\supseteq}$ and $\mathcal{P}_1^{\supseteq}$ before considering any clauses as possible members of $\mathcal{U}$. The details are omitted here, but the result is avoidance of unnecessary subsumption checks. Furthermore, keeping $\mathcal{P}_0^{\supseteq}$ and $\mathcal{P}_0^{\bowtie}$ as separate sets makes prime marks at the ends of trie branches unnecessary; checking for them in the original algorithm is almost as expensive as the subsumption check itself. Most significantly, clause set operations can be realized recursively on entire sets, represented as tries.[1] Experimentally, the advantage increases with the size of the trie.

[1] Tries have been employed for (even first order) subsumption [19], but on a clause to trie basis, rather than the trie to trie basis developed here.

The operators $\mathit{Subsumed}(F, G) = \{C \in G \mid C \text{ is subsumed by some } C' \in F\}$ and $\mathit{XUnions}(F, G) = \{C \cup D \mid C \in F,\ D \in G,\ C \cup D \text{ is not tautological }\}$ on clause sets F and G are defined as set operations, but the pseudocode implements them on clause sets represented as ternary tries. Recall that the trie $\mathcal{T}$ can be written $\langle r, \mathcal{T}^+, \mathcal{T}^-, \mathcal{T}^0 \rangle$, where r is the root label of $\mathcal{T}$, and $\mathcal{T}^+$, $\mathcal{T}^-$, and $\mathcal{T}^0$ are the three subtries. Tries with three empty children are called *leaves*.

input : Two clausal tries T_1 and T_2
output: T, a trie containing all the clauses in T_2 subsumed by some clause in T_1
if $T_1 = null$ *or* $T_2 = null$ **then**
  $T \leftarrow null$;
else if *leaf*(T_1) **then**
  $T \leftarrow T_2$;
else
  $T \leftarrow$ new Leaf;
  $T^+ \leftarrow \mathit{Subsumed}(T_1^+, T_2^+) \cup \mathit{Subsumed}(T_1^0, T_2^+)$;
  $T^- \leftarrow \mathit{Subsumed}(T_1^-, T_2^-) \cup \mathit{Subsumed}(T_1^0, T_2^-)$;
  $T^0 \leftarrow \mathit{Subsumed}(T_1^0, T_2^0)$;
  if *leaf*(T) **then**
    $T \leftarrow null$;
  end
end
return T;

Algorithm 1. *Subsumed*(T_1,T_2)

input : Two clausal tries T_1 and T_2
output: T, a trie of the pairwise unions of the clauses in T_1 and T_2
if $T_1 = null$ *or* $T_2 = null$ **then**
  $T \leftarrow null$;
else if *leaf*(T_1) **then**
  $T \leftarrow T_2$;
else if *leaf*(T_2) **then**
  $T \leftarrow T_1$;
else
  $T \leftarrow$ new Leaf;
  $T^+ \leftarrow \mathit{XUnions}(T_1^+, T_2^+) \cup \mathit{XUnions}(T_1^0, T_2^+) \cup \mathit{XUnions}(T_1^+, T_2^0)$;
  $T^- \leftarrow \mathit{XUnions}(T_1^-, T_2^-) \cup \mathit{XUnions}(T_1^0, T_2^-) \cup \mathit{XUnions}(T_1^-, T_2^0)$;
  $T^0 \leftarrow \mathit{XUnions}(T_1^0, T_2^0)$;
  if *leaf*(T) **then**
    $T \leftarrow null$;
  end
end
return T;

Algorithm 2. *XUnions*(T_1,T_2)

Experiments involving subsumption testing are reported below. In Section 5, the new and original versions of the pi-trie algorithms are compared.

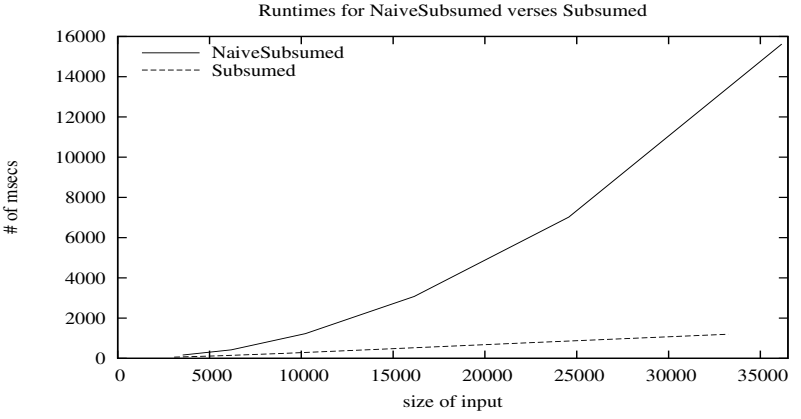


Fig. 1. *Subsumed* vs. *NaiveSubsumed*

The input for the experiments depicted in Figure 1 is a pair of n-variable CNF formulas, where $n \in \{10, \ldots, 15\}$ with results averaged over 20 trials for each n. Each formula with n variables has $\lfloor \binom{n}{3}/4 \rfloor$ clauses of length 3, $\lfloor \binom{n}{4}/2 \rfloor$ clauses of length 4, and $\binom{n}{5}$ clauses of length 5. This corresponds to $\frac{1}{32}$ of the $2^k\binom{n}{k}$ possible clauses of length k for $k = 3, 4, 5$.

The two clause sets are compiled into two tries for the application of *Subsumed* and into two lists for the application of *NaiveSubsumed*. (The latter is a straightforward subset test on lists and is not shown.) The ratio of the runtimes changes as the input size increases, suggesting that the runtimes of *NaiveSubsumed* and *Subsumed* differ asymptotically. Additional evidence is supplied by Lemma 3:

Lemma 3. *Subsumed*, when applied to two full ternary tries of depth h and combined size $n = 2(\frac{3^{h+1}-1}{2})$, runs in time $O(n^{\frac{\log 5}{\log 3}}) \approx O(n^{1.465})$. □

This is less than *NaiveSubsumed*'s obvious runtime of $O(n^2)$ but still more than linear. Lemma 3 is interesting but the general upper bound may be quite different.

5 The New *pi*-trie Algorithm

Theorem 2 provides a simpler characterization of the *pi*-trie algorithm than the one developed in [12]. The algorithm can be viewed in the standard divide-and-conquer framework,[2] where each problem $\mathcal{F}$ is divided into subproblems $\mathcal{F}_0, \mathcal{F}_1$ by substitution on the appropriate variable. The base case is when substitution yields a constant, so that $\mathcal{P}(0) = \{\{\}\}$ or $\mathcal{P}(1) = \{\}$.

The remainder of the algorithm consists of combining $\mathcal{P}_0$ and $\mathcal{P}_1$ to form $\mathcal{P}(\mathcal{F})$. This is done both here and in [12] by a routine called PIT. However, in [12], the subsumption

[2] Rudell [17] proposed a similar divide-and-conquer strategy for prime implicants absent of any analysis of precisely when and how subsumption checks are applied.

input : A boolean formula $\mathcal{F}$ and a list of its variables $V = \langle v_1, \ldots, v_k \rangle$
output: The clause set $\mathcal{P}(\mathcal{F})$ — the prime implicates of $\mathcal{F}$
if $\mathcal{F} = 1$ **then**
    return $\emptyset$; `// Tautologies have no prime implicates.`
else if $\mathcal{F} = 0$ **then**
    return $\{\{\}\}$; `//` $\mathcal{P}(0)$ `is the set of just the empty clause.`
else
    $\mathcal{F}_0 \leftarrow \mathcal{F}[0/v_1]$;
    $\mathcal{F}_1 \leftarrow \mathcal{F}[1/v_1]$;
    $V' \leftarrow \langle v_2, \ldots, v_k \rangle$;
    return PIT(prime($\mathcal{F}_0, V'$) , prime($\mathcal{F}_1, V'$) , v_1);
end

Algorithm 3. prime($\mathcal{F}$,V)

input : Clause sets $\mathcal{P}_0 = \mathcal{P}(\mathcal{F}_0)$ and $\mathcal{P}_1 = \mathcal{P}(\mathcal{F}_1)$, variable v
output: The clause set $\mathcal{P} = \mathcal{P}((v \vee \mathcal{F}_0) \wedge (\neg v \vee \mathcal{F}_1))$
$\mathcal{P}_0^{\supseteq} \leftarrow$ *Subsumed*$(\mathcal{P}_1, \mathcal{P}_0)$; `// Initialize` $\mathcal{P}_0^{\supseteq}$
$\mathcal{P}_1^{\supseteq} \leftarrow$ *Subsumed*$(\mathcal{P}_0, \mathcal{P}_1)$; `// Initialize` $\mathcal{P}_0^{\supseteq}$
$\mathcal{P}_0^{\bowtie} \leftarrow \mathcal{P}_0 - \mathcal{P}_0^{\supseteq}$;
$\mathcal{P}_1^{\bowtie} \leftarrow \mathcal{P}_1 - \mathcal{P}_1^{\supseteq}$;
$\mathcal{U} \leftarrow$ *XUnions*$(\mathcal{P}_0^{\bowtie}, \mathcal{P}_1^{\bowtie})$;
$\mathcal{U} \leftarrow \mathcal{U} -$ *SubsumedStrict*$(\mathcal{U},\mathcal{U})$;
$\mathcal{U} \leftarrow \mathcal{U} -$ *Subsumed*$(\mathcal{P}_0^{\supseteq},\mathcal{U})$;
$\mathcal{U} \leftarrow \mathcal{U} -$ *Subsumed*$(\mathcal{P}_1^{\supseteq},\mathcal{U})$;
return $((v \vee \mathcal{P}_0^{\bowtie}) \cup (\neg v \vee \mathcal{P}_1^{\bowtie}) \cup \mathcal{P}_0^{\supseteq} \cup \mathcal{P}_1^{\supseteq} \cup \mathcal{U})$;

Algorithm 4. PIT($\mathcal{P}_0$,$\mathcal{P}_1$,v)

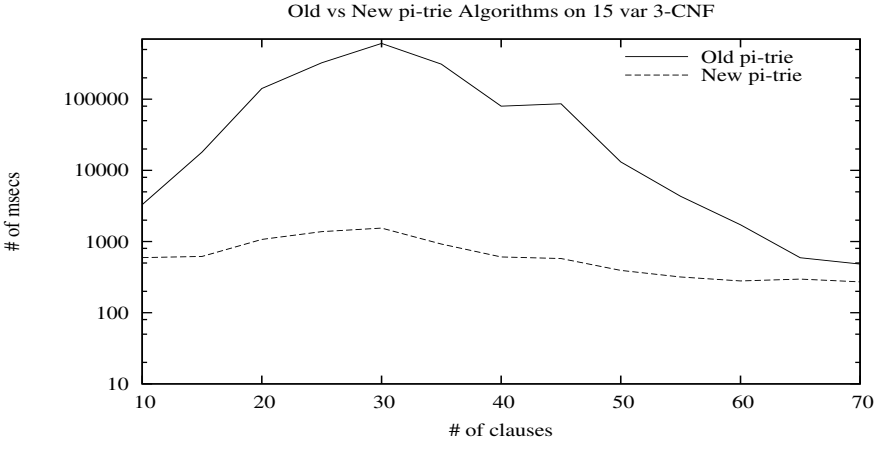


Fig. 2. Old vs. New *pi*-trie algorithm

checking is done branch by branch; here, it is performed between entire tries denoted by $\mathcal{P}_0$, $\mathcal{P}_1$, $\mathcal{P}_0^{\supseteq}$, $\mathcal{P}_0^{\bowtie}$, $\mathcal{P}_1^{\supseteq}$, and $\mathcal{P}_1^{\bowtie}$. Trie-based operations are employed such as $Subsumed$ and $Unions$, which were discussed in Section 4.

Figure 2 compares the *pi*-trie algorithm from [12] to the updated version using the recursive $Subsumed$ and $XUnion$ operators. The input for both algorithms is 15-variable 3-CNF with varying numbers of clauses, and the runtimes are averaged over 20 trials. The great discrepancy between runtimes requires that they be presented in log scale; it is explained in part by Figure 1, which compares the runtime of *Subsumed* to Algorithm 5, a naïve subsumption algorithm. The performance of the two systems converges as the number of clauses increases. With more clauses, formulas are unsatisfiable with probability approaching 1. As a result, the base cases of the prime algorithm are encountered early, and subsumption in the PIT routine plays a less dominant role, diminishing the advantage of the new algorithm.

6 Selective Computation

It is sometimes useful to restrict attention to a subset of prime implicates having some particular property, such as being positive or having length at most four. The subset can always be selected from the entire set of prime implicates, but generating only the prime implicates of interest is preferable (if this improves efficiency!). If Q is a property that sets may or may not have, and if A and B are sets, then Q is said to be *superset invariant* if whenever $B \supseteq A$, $Q(A) \rightarrow Q(B)$; it is *subset invariant* if whenever $B \subseteq A$, $Q(A) \rightarrow Q(B)$. The complement property, denoted $\overline{Q}$, is defined in the obvious way: $\overline{Q}(X) = \neg Q(X)$. Examples of subset invariant properties of clauses are *containing no more than three literals*, *containing only positive literals*, and *being a Horn clause*. The following lemma is immediate.

Lemma 4. If Q is superset (subset) invariant, then $\overline{Q}$ is subset (superset) invariant. □

It turns out that the pi-trie algorithm is particularly amenable to being tuned to generate only prime implicates satisfying subset invariant properties. The reason is that clauses computed at any stage of the algorithm always contain as subsets clauses computed at an earlier stage.

The main function of the pi-trie algorithm is **prime**, which returns the set (represented as a trie) of prime implicates of its first argument, $\mathcal{F}$. This set is simply the result of the PIT routine called on the prime implicates of $\mathcal{F}_0$ and $\mathcal{F}_1$ as actual parameters and represented in PIT by the formal parameters $\mathcal{P}_0$ and $\mathcal{P}_1$. Examining the assignment statements in PIT, it is easily verified that $\mathcal{P}_0^{\bowtie}$ and $\mathcal{P}_1^{\bowtie}$ consist of subsets of $\mathcal{P}_0$ and $\mathcal{P}_1$. Furthermore, every clause initially in $\mathcal{U}$ is the union of two clauses from $\mathcal{P}_0^{\bowtie}$ and $\mathcal{P}_1^{\bowtie}$, and hence from $\mathcal{P}_0$ and $\mathcal{P}_1$. Subsequent modifications of $\mathcal{U}$ are set subtractions. Therefore, in the return statement, each clause of $v \vee \mathcal{P}_0^{\bowtie}$, $\neg v \vee \mathcal{P}_1^{\bowtie}$, $\mathcal{P}_0^{\supseteq} \cup \mathcal{P}_1^{\supseteq}$, and $\mathcal{U}$ may contain v, $\neg v$, or neither, and otherwise consists of exactly one clause from either $\mathcal{P}_0$ or $\mathcal{P}_1$, or of exactly two clauses – one each from $\mathcal{P}_0$ and $\mathcal{P}_1$. This proves

Lemma 5. Let $C \in$ PIT(prime($\mathcal{F}_0, V'$) , prime($\mathcal{F}_1, V'$), v_1) and let $C_0 \in$ prime($\mathcal{F}_0, V'$). Then either $C \cap C_0 = \emptyset$ or $C \cap C_0 = C_0$. Similarly for $C_1 \in$ prime($\mathcal{F}_1, V'$). □

Theorem 3. Let Q be a subset invariant property, and let $\mathcal{F}$ be a propositional formula. Construct PIT^{Q} to be PIT modified to first delete all clauses in $\mathcal{P}_0$ and in $\mathcal{P}_1$ having property $\overline{Q}$. Let $S = \mathrm{prime}(\mathcal{F}_0, V') \cup \mathrm{prime}(\mathcal{F}_1, V')$. Partition S into the prime implicates having property Q, denoted S_Q, and those having property $\overline{Q}$, denoted $S_{\overline{Q}}$. Then the clauses produced by PIT^{Q}(prime($\mathcal{F}_0, V'$) , prime($\mathcal{F}_1, V'$), v_1) include all those produced by PIT(prime($\mathcal{F}_0, V'$) , prime($\mathcal{F}_1, V'$), v_1) having property Q. □

Observe that the theorem does not guarantee that PIT returns *only* clauses of S_Q. By adding a literal to clauses in $\mathcal{P}_0^{\bowtie}$ or in $\mathcal{P}_1^{\bowtie}$, or by forming a union of clauses with Q, a clause with property $\overline{Q}$ may result (e.g., when Q is a length restriction.)

The point is that clauses with $\overline{Q}$ are not required to generate clauses with Q. Since PIT^{Q} is invoked recursively by prime, each invocation is pruning only the most recently generated clauses with $\overline{Q}$. Therefore, the clauses returned by the initial invocation of prime should be pruned one final time, but this is an implementation detail.

Restricted sets of prime implicates are useful in several settings. One is abductive reasoning: Finding short non-redundant explanations will result from restricting the length of prime implicates of the theory.

Perhaps surprisingly, prime implicates turn out to be useful in the synthesis of correct code for systems having *polychronous data flow* (both synchronous and asynchronous data flow). In [8], a *calculus of epochs* is developed which requires computations of prime implicates. The most useful in this setting are positive — no negative literals present — and short — containing only a few literals. Positive, unit prime implicates are particularly useful. Of course, these are all subset invariant properties.

7 Experiments with Subset Invariant Properties

The new pi-trie algorithm — with subset invariant filters for maximum clause length and for designated forbidden literals — is implemented in Java. Its performance has been compared with two others.[3] The first comes from a system of Zhuang, Pagnucco and Meyer [22]. The second comparison was made with a somewhat simpler resolution-based prime implicate generator called **ResPrime**. Also implemented in Java, it exhaustively iterates resolution and subsumption checking.

The input for all experiments is a variable number of random 3-CNF clauses from a constant alphabet size. For a good experimental analysis of prime implicates in random 3-CNF, see [18]. The algorithm from [22] proved to be much slower than the others and could not be illustrated in Figure 3, which shows a comparison of runtimes of **ResPrime**, pi-trie, and pi-trie filtered to exclude clauses of length greater than 2.

While **ResPrime** outperforms the pi-trie algorithm on small input, the reverse is true on larger input. When the number of clauses is small, the number of possible resolutions on small clause sets is correspondingly small, whereas the pi-trie algorithm does not assume the input to be CNF and thus does not take advantage of this syntactic feature. Probabilistically, as the number of clauses increases, the formulas become less and less satisfiable, and so in general the recursive substitution that drives the pi-trie algorithm requires fewer substitutions to reach a contradiction, allowing faster runtimes. Much

[3] It is surprisingly difficult to find publicly available prime implicate generators.

of the work so avoided is also avoided by filtering, and this lessens the advantage of filtering itself with formulas whose size is large compared with the number of variables (see Figure 4 below). At the same time, the resolution algorithm is required to process more and more clauses and their resolution pairs, and so runs slower.

The pi-trie algorithm with filtering offers dramatic improvements in performance on searches for subset invariant prime implicates. Figure 4 has the results of an experiment with a 13-variable 3-CNF formula. Two filters are used: The first is "max length 2," and the second excludes clauses containing any of the literals v_3, v_5, v_6, or $\neg v_7$.

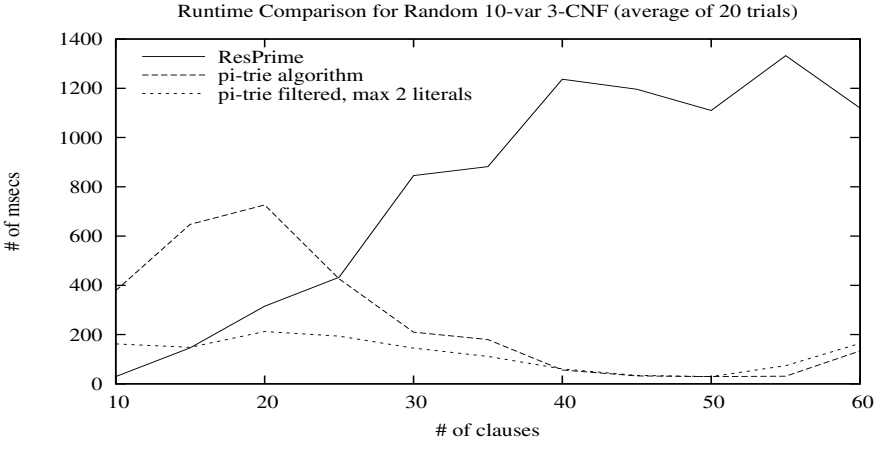


Fig. 3. 10 var 3-CNF

Note that especially for the length-limited filter, significant efficiency gains can be attained by applying the filter during generation as opposed to generating all prime implicates and then filtering, as most other algorithms require. In fact, besides [11], which uses a novel integer linear programming approach to discover small prime implicants (easily transformed to a prime implicate algorithm via duality), the authors are not aware of any other algorithms that allow filtering during generation of prime implicates based on size or on designated literals.

The results described in Figure 4 for formulas having 20 clauses are noteworthy: The raw pi-trie algorithm averaged 22,430 msecs, whereas the "max length 2" filter averaged only 1,170 msecs. One reason for the runtime advantage is that 13-variable 20-clause formulas have relatively long prime implicates. The average number of prime implicates is 213.3, while the average number with length at most 2 is 0.7.

Generating all prime implicates is considerably more difficult than deciding satisfiability — the number of prime implicates of a formula, and thus the size of the output, is typically exponential [3]. These experiments were thus performed on relatively small input compared to the examples that modern DPLL-based SAT solvers can handle.

Suppose now that a filter is applied that prunes all prime implicates having length greater than some fixed k. If n is the number of variables, the number of prime

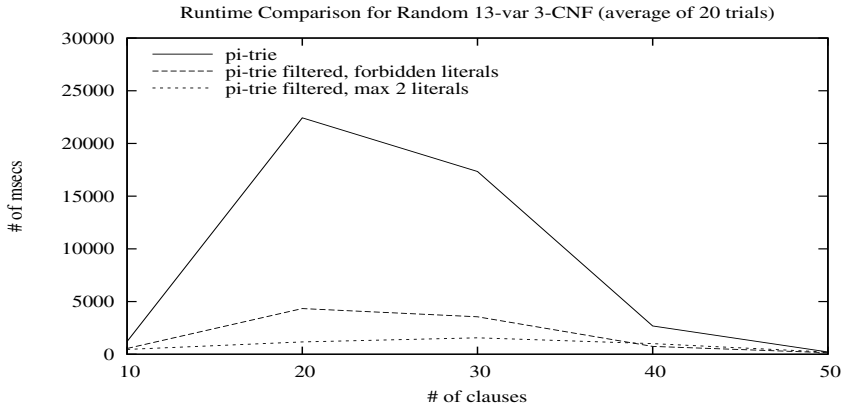


Fig. 4. *pi*-trie Filtering

implicates of length k or less is bounded above by $\binom{n}{k} \cdot 3^k$ which is polynomial in n of degree at most k. This proves.

Theorem 4. Let $\mathcal{F}$ be a formula with n variables, and let Q be a subset invariant property satisfied only by clauses of length k or less. Then if the pi-trie algorithm employs PIT^Q, it runs in polynomial space. □

Placing an upper bound on length or restricting prime implicates to consist only of certain designated literals will therefore cause the pi-trie algorithm to run in polynomial space. Note that for any one problem instance, a restriction to designated literals amounts to prohibiting the complement set of literals. But if the problem is parameterized on the size of the complement set, polynomial space computation does not result because for a fixed k, as n increases, so does admissible clause length.

Reference

1. Bittencourt, G.: Combining syntax and semantics through prime form representation. Journal of Logic and Computation 18, 13–33 (2008)
2. Castell, T.: Computation of prime implicates and prime implicants by a variant of the davis and putnam procedure. In: ICTAI, pp. 428–429 (1996)
3. Chandra, A., Markowsky, G.: On the number of prime implicants. Discrete Mathematics 24, 7–11 (1978)
4. Coudert, O., Madre, J.: Implicit and incremental computation of primes and essential implicant primes of boolean functions. In: 29th ACM/IEEE Design Automation Conference, pp. 36–39 (1992)
5. de Kleer, J.: An improved incremental algorithm for computing prime implicants. In: Proc. AAAI 1992, San Jose, CA, pp. 780–785 (1992)
6. Jackson, P.: Computing prime implicants incrementally. In: Kapur, D. (ed.) CADE 1992. LNCS(LNAI), vol. 607, pp. 253–267. Springer, Heidelberg (1992)
7. Jackson, P., Pais, J.: Computing prime implicants. In: Stickel, M.E. (ed.) CADE 1990. LNCS(LNAI), vol. 449, pp. 543–557. Springer, Heidelberg (1990)

8. Jose, B.A., Shukla, S.K., Patel, H.D., Talpin, J.P.: On the deterministic multi-threaded software synthesis from polychronous specifications. In: Formal Models and Methods in Co-Design (MEMOCODE 2008), Anaheim, California (June 2008)
9. Kean, A., Tsiknis, G.: An incremental method for generating prime implicants/implicates. Journal of Symbolic Computation 9, 185–206 (1990)
10. Manquinho, V.M., Flores, P.F., Silva, J.P.M., Oliveira, A.L.: Prime implicant computation using satisfiability algorithms. In: Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, Newport Beach, U.S.A., pp. 232–239 (November 1997)
11. Marques-Silva, J.P.: On computing minimum size prime implicants. In: International Workshop on Logic Synthesis (1997)
12. Matusiewicz, A., Murray, N.V., Rosenthal, E.: Prime implicate tries. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS(LNAI), vol. 5607, pp. 250–264. Springer, Heidelberg (2009)
13. Matusiewicz, A., Murray, N.V., Rosenthal, E.: Tri-based subsumption and selective computation of prime implicates. Technical Report SUNYA-CS-11-01. Department of Computer Science, University at Albany - SUNY (March 2011)
14. Ngair, T.: A new algorithm for incremental prime implicate generation. In: Proc. IJCAI 1993, Chambery, France (1993)
15. Quine, W.V.: The problem of simplifying truth functions. The American Mathematical Monthly 59(8), 521–531 (1952)
16. Ramesh, A., Becker, G., Murray, N.V.: Cnf and dnf considered harmful for computing prime implicants/implicates. Journal of Automated Reasoning 18(3), 337–356 (1997)
17. Rudell, R.L.: Logic Synthesis for VLSI Design. In: PhD thesis, EECS Department. University of California, Berkeley (1989)
18. Schrag, R., Crawford, J.M.: Implicates and prime implicates in random 3-SAT. Artificial Intelligence 81(1-2), 199–222 (1996)
19. Schulz, S.: Simple and efficient clause subsumption with feature vector indexing. In: Basin, D., Rusinowitch, M. (eds.) IJCAR 2004. LNCS (LNAI), vol. 3097. Springer, Heidelberg (2004)
20. Slagle, J.R., Chang, C.L., Lee, R.C.T.: A new algorithm for generating prime implicants. IEEE Transactions on Computers 19(4), 304–310 (1970)
21. Strzemecki, T.: Polynomial-time algorithm for generation of prime implicants. Journal of Complexity 8, 37–63 (1992)
22. Zhuang, Z.Q., Pagnucco, M., Meyer, T.: Implementing iterated belief change via prime implicates. In: Australian Conference on Artificial Intelligence, pp. 507–518 (2007)

Cholesky Decomposition Rectification for Non-negative Matrix Factorization

Tetsuya Yoshida

Graduate School of Information Science and Technology,
Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
yoshida@meme.hokudai.ac.jp

Abstract. We propose a method based on Cholesky decomposition for Non-negative Matrix Factorization (NMF). NMF enables to learn local representation due to its non-negative constraint. However, when utilizing NMF as a representation leaning method, the issues due to the non-orthogonality of the learned representation has not been dealt with. Since NMF learns both feature vectors and data vectors in the feature space, the proposed method 1) estimates the metric in the feature space based on the learned feature vectors, 2) applies Cholesky decomposition on the metric and identifies the upper triangular matrix, 3) and utilizes the upper triangular matrix as a linear mapping for the data vectors. The proposed approach is evaluated over several real world datasets. The results indicate that it is effective and improves performance.

1 Introduction

Previous representation learning methods have not explicitly considered the characteristics of algorithms applied to the learned representation [4]. When applying Non-negative Matrix Factorization (NMF) [5,6,8,1] to document clustering, in most cases the number of features are set to the number of clusters [8,2]. However, when the number of features is increased, the non-orthogonality of the features in NMF hinder the effective utilization of the learned representation.

We propose a method based on Cholesky decomposition [3] to remedy the problem due to the non-orthogonality of features learned in NMF. Since NMF learns both feature vectors and data vectors in the feature space, the proposed method 1) first estimates the metric in the feature space based on the learned feature vectors, 2) applies Cholesky decomposition on the metric and identifies the upper triangular matrix, 3) and finally utilize the upper triangular matrix as a linear mapping for the data vectors.

The proposed method is evaluated over several document clustering problem, and the results indicate the effectiveness of the proposed method. Especially, the proposed method enables the effective utilization of the the learned representation by NMF without modifying the algorithms applied to the learned representation. No label information is required to exploit the metric in the feature space, and the proposed method is fast and robust, since Cholesky decomposition is utilized [3].

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 214–219, 2011.

2 Cholesky Decomposition Rectification for NMF

We use a bold capital letter for a matrix, and a lower italic letter for a vector. $\mathbf{X}_{ij}$ stands for the element in a matrix $\mathbf{X}$. tr stands for the trace of a matrix, and $\mathbf{X}^T$ stands for the transposition of $\mathbf{X}$.

2.1 Non-negative Matrix Factorization

Under the specified number of features q, Non-negative Matrix Factorization (NMF) [6] factorizes a non-negative matrix $\mathbf{X}$= [$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$] $\in \mathbb{R}_+^{p\times n}$ into two non-negative matrices $\mathbf{U}$=[$\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q$] $\in \mathbb{R}_+^{p\times q}$, $\mathbf{V}$=[$\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$] $\in \mathbb{R}_+^{q\times n}$ as

$$\mathbf{X} \simeq \mathbf{UV} \tag{1}$$

Each $\boldsymbol{x}_i$ is approximated as a linear combination of $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q$. Minimization of the following objective function is conducted to obtain the matrices $\mathbf{U}$ and $\mathbf{V}$:

$$J_0 = ||\mathbf{X} - \mathbf{UV}||^2 \tag{2}$$

where $||\cdot||$ stands for the norm of a matrix. In this paper we focus on Frobenius norm $||\cdot||_F$ [6]. Compared with methods based on eigenvalue analysis such as PCA, each element of $\mathbf{U}$ and $\mathbf{V}$ are non-negative, and their column vectors are not necessarily orthogonal in Euclidian space.

2.2 Clustering with NMF

Besides image analysis [5], NMF is also applied to document clustering [8,2]. In most approaches which utilize NMF for document clustering, the number of features are set to the number of clusters [8,2]. Each instance is assigned to the cluster c with the maximal value in the constructed representation $\boldsymbol{v}$.

$$c = \underset{c}{\operatorname{argmax}}\, v_c \tag{3}$$

where v_c stands for the value of c-th element in $\boldsymbol{v}$.

2.3 Representation Learning with NMF

When NMF is considered as a dimensionality reduction method, some learning method such as SVM (Support Vector Machine) or kmeans is applied for the learned representation $\mathbf{V}$. In many cases, methods which assume Euclidian space (such as kmeans) are utilized for conducting learning on $\mathbf{V}$ [4]. However, to the best of our knowledge, the issues arising from the non-orthogonality of the learned representation has not been dealt with.

2.4 Cholesky Decomposition Rectification

One of the reasons of the above problem is that, when the learned representation $\mathbf{V}$ is utilized, usually the square distance between a pair of instances ($\boldsymbol{v}_i$, $\boldsymbol{v}_j$) is calculated as $(\boldsymbol{v}_i\text{-}\ \boldsymbol{v}_j)^T(\boldsymbol{v}_i\text{-}\ \boldsymbol{v}_j)$ by (implicitly) assuming that $\boldsymbol{v}_i$ is represented in

some Euclidian space. However, since $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q$ learned by NMF are not orthogonal each other in general, the above calculation is not appropriate when NMF is utilized to learn $\mathbf{V}$. If we know the metric $\mathbf{M}$ which reflects non-orthogonality in the feature space, the square distance can be calculated as

$$(\boldsymbol{v}_i - \boldsymbol{v}_j)^T \mathbf{M} (\boldsymbol{v}_i - \boldsymbol{v}_j) \tag{4}$$

This corresponds to the (squared) Mahalanobis generalized distance.

We exploit the property of NMF in the sense that data matrix $\mathbf{X}$ is decomposed into i) $\mathbf{U}$, whose column vectors spans the feature space, and ii) $\mathbf{V}$, which are the representation in the feature space. Based on this property, the proposed method 1) first estimates the metric in the feature space based on the learned feature vectors, 2) applies Cholesky decomposition on the metric and identifies the upper triangular matrix, 3) and finally utilizes the upper triangular matrix as a linear mapping for the data vectors. Some learning algorithm is applied to the transformed representation from 3) as in [4],

We explain 1) and 2) in our proposed method. Note that the proposed method enables to effectively utilize the learned representation by NMF, without modifying the algorithms applied to the learned representation.

Estimation of Metric via NMF. In NMF, when approximating the data matrix $\mathbf{X}$ and representing it as $\mathbf{V}$ in the feature space, the explicit representation of features in the original data space can also be obtained as $\mathbf{U}$. Thus, by normalizing each $\boldsymbol{u}$ such that $\boldsymbol{u}^T\boldsymbol{u}$=1 as in [8], we estimate the metric $\mathbf{M}$ as the Gram matrix $\mathbf{U}^T\mathbf{U}$ of the features.

$$\mathbf{M} = \mathbf{U}^T\mathbf{U}, \quad s.t. \;\; \boldsymbol{u}_l^T \boldsymbol{u}_l = 1, \quad \forall l = 1, \ldots, q \tag{5}$$

Contrary to other metric learning approaches, no label information is required to estimate $\mathbf{M}$ in our approach. Furthermore, since each data is approximated (embedded) in the feature space spanned by $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q$, it seems rather natural to utilize eq.(5) based on $\mathbf{U}$ to estimate the metric of the feature space.

Cholesky Decomposition Rectification. Since the metric $\mathbf{M}$ is estimated by eq.(5), it is guaranteed that $\mathbf{M}$ is symmetric positive semi-definite. Thus, based on Linear algebra [3], it is guaranteed that $\mathbf{M}$ can be *uniquely* decomposed by Cholesky decomposition with the upper triangular matrix $\mathbf{T}$ as:

$$\mathbf{M} = \mathbf{T}^T\mathbf{T} \tag{6}$$

By substituting eq.(6) into eq.(4), we obtain the rectified representation $\mathbf{TV}$:

$$\mathbf{V} \rightarrow \mathbf{TV} \tag{7}$$

based on the upper triangular matrix $\mathbf{T}$ via Cholesky decomposition.

Algorithm 1. Cholesky Decomposition Rectification for NMF (CNMF)

CNMF(X, $algNMF$, q, $parameters$)
Require: $\mathbf{X} \in \mathbb{R}_{+}^{p \times n}$ *//data matrix*
Require: $algNMF$; *//the utilized NMF algorithm*
Require: q; *//the number of features*
Require: $pars$; *//other parameters in* $algNMF$
1: $\mathbf{U}, \mathbf{V} :=$ run $algNMF$ on $\mathbf{X}$ with q (and $pars$) s.t. $\boldsymbol{u}_l^T \boldsymbol{u}_l = 1, \forall l = 1, \ldots, q$
2: $\mathbf{M} := \mathbf{U}^T\mathbf{U}$
3: $\mathbf{T} :=$ Cholesky decomposition of $\mathbf{M}$ s.t. $\mathbf{M} = \mathbf{T}^T\mathbf{T}$
4: **return** $\mathbf{U}$, $\mathbf{TV}$

The proposed algorithm CNMF is shown in Algorithm 1.

3 Evaluations

3.1 Experimental Settings

Datasets. We evaluated the proposed algorithm on 20 Newsgroup data (20NG)[1]. Each document is represented as the standard vector space model based on the occurrences of terms. We created three datasets for 20NG (Multi5, Multi10, Multi15 datasets, with 5, 10, 15 clusters). 50 documents were sampled from each group (cluster) in order to create a sample for one dataset, and 10 samples were created for each dataset. For each sample, we conducted stemming using porter stemmer[2] and MontyTagger[3], removed stop words, and selected 2,000 words with large mutual information. We conducted experiments on the TREC datasets, however, results on other datasets are omitted due to page limit.

Evaluation Measures. For each dataset, the cluster assignment was evaluated with respect to Normalized Mutual Information (NMI). Let C, $\hat{C}$ stand for the random variables over the true and assigned clusters. NMI is defined as $NMI = \frac{I(\hat{C};C)}{(H(\hat{C})+H(C))/2} (\in [0,1])$ where $H(\cdot)$ is Shannon Entropy, $I(;)$ is Mutual Information. NMI corresponds to the accuracy of assignment. The larger NMI is, the better the result is.

Comparison. We utilized the proposed method on 1) NMF [6], 2) WNMF [8], 3) GNMF [1], and evaluated its effectiveness. Since these methods are partitioning based clustering methods, we assume that the number of clusters k is specified.

WNMF [6] first converts the data matrix $\mathbf{X}$ utilizing the weighting scheme in Ncut [7], and applies the standard NMF algorithm on the converted data.

GNMF [1] constructs the m nearest neighbor graph and utilizes the graph Laplacian for the adjacency matrix $\mathbf{A}$ of the graph as a regularization term as:

[1] http://people.csail.mit.edu/ jrennie/20Newsgroups/. 20news-18828 was utilized.
[2] http://www.tartarus.org/ martin/PorterStemmer
[3] http://web.media.mit.edu/ hugo/montytagger

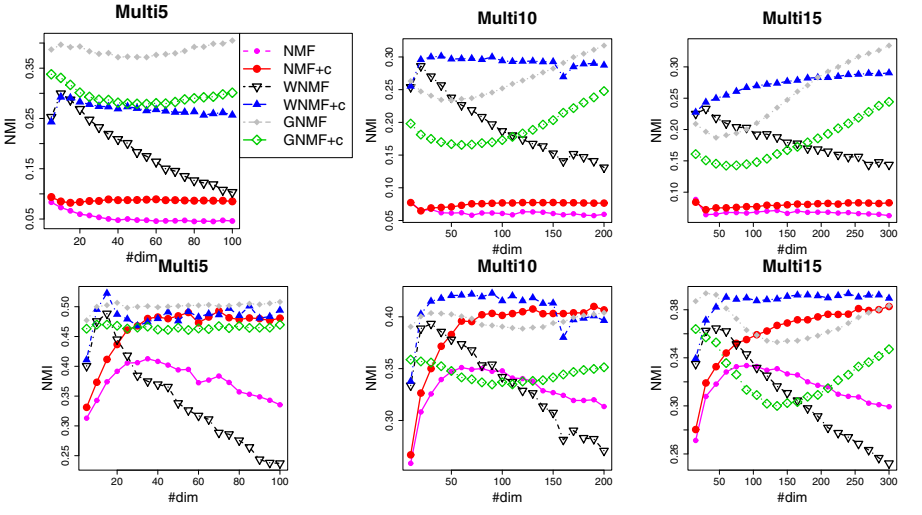


Fig. 1. Results on 20 Newsgroup datasets (*NMI*) (upper:kmeansClower:skmeans)

$$J_2 = ||\mathbf{X} - \mathbf{UV}||^2 + \lambda \operatorname{tr}(\mathbf{VLV}^T) \tag{8}$$

where $\mathbf{L} = \mathbf{D}$ - $\mathbf{A}$ ($\mathbf{D}$ is the degree matrix), and λ is the regularization parameter.

Parameters. Cosine similarity, was utilized as the pairwise similarity measure. We varied the value of q and conducted experiments. The number of neighbors m was set to 10 in GNMF, and λ was set to 100 based on [1]. The number of maximum iterations was set to 30.

Evaluation Procedure. As the standard clustering methods based on Euclidian space, kmeans and skmeans were applied to the learned representation matrix $\mathbf{V}$ from each method, and the proposed representation $\mathbf{TV}$ in eq.(7).

Since NMF finds out local optimal, the results ($\mathbf{U}$, $\mathbf{V}$) depends on the initialization. Thus, we conducted 10 random initialization for the same data matrix. Furthermore, since both kmeans and skmeans are affected from the initial cluster assignment, for the same representation (either $\mathbf{V}$ or $\mathbf{TV}$), clustering was repeated 10 times with random initial assignment.

3.2 Results

The reported figures are the average of 10 samples in each dataset[4]. The horizontal axis corresponds to the number of features q, the vertical one to *NMI*. In the legend, solid lines correspond to NMF, dotted lines to WNMF, and dash lines to GNMF. In addition, +c stands for the results by utilizing the proposed method in eq.(7) and constructing $\mathbf{TV}$ for each method.

[4] The average of 1,000 runs is reported for each dataset.

The results in Fig. 1 show that the proposed method improves the performance of kmeans (the standard Euclidian distance) and skmeans (cosine similarity in Euclidian space). Thus, the proposed method can be said as effective to improve the performance. Especially, skmeans was substantially improved (lower figures in Fig. 1). In addition, when the proposed method is applied to WNMF (blue dotted WNMF+c), equivalent or even better performance was obtained compared with GNMF. On the other hand, the proposed method was not effective to GNMF, since the presupposition in Section 2.4 does not hold in GNMF.

As the number of features q increases, the performance of NMF and WNMF degraded. On the other hand, by utilizing the proposed method, NMF+c and WNMF+c were very robust with respect to the increase of q. Thus, the proposed method can be said as effective for utilizing large number of features in NMF.

4 Concluding Remarks

We proposed a method based on Cholesky decomposition to remedy the problem due to the non-orthogonality of features learned in Non-negative Matrix Factorization (NMF). Since NMF learns both feature vectors and data vectors in the feature space, the proposed method 1) first estimates the metric in the feature space based on the learned feature vectors, 2) applies Cholesky decomposition on the metric and identifies the upper triangular matrix, 3) and finally utilize the upper triangular matrix as a linear mapping for the data vectors. The proposed method enables the effective utilization of the learned representation by NMF without modifying the algorithms applied to the learned representation.

References

1. Cai, D., He, X., Wu, X., Han, J.: Non-negative matrix factorization on manifold. In: Proc. of ICDM 2008, pp. 63–72 (2008)
2. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: Proc. of KDD 2006, pp. 126–135 (2006)
3. Harville, D.A.: Matrix Algebra From a Statistican's Perspective. Springer, Heidelberg (2008)
4. Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: Proc. of IJCAI 2003, pp. 561–566 (2003)
5. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
6. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Proc. of Neural Information Processing Systems (NIPS), pp. 556–562 (2001)
7. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
8. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proc. of SIGIR 2003, pp. 267–273 (2003)

A New Method for Adaptive Sequential Sampling for Learning and Parameter Estimation

Jianhua Chen[1] and Xinjia Chen[2]

[1] Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803-4020
jianhua@csc.lsu.edu
[2] Department of Electrical Engineering
Southern University
Baton Rouge, LA 70813
xinjiachen@engr.subr.edu

Abstract. Sampling is an important technique for parameter estimation and hypothesis testing widely used in statistical analysis, machine learning and knowledge discovery. In contrast to batch sampling methods in which the number of samples is known in advance, adaptive sequential sampling gets samples one by one in an on-line fashion without a pre-defined sample size. The stopping condition in such adaptive sampling scheme is dynamically determined by the random samples seen so far. In this paper, we present a new adaptive sequential sampling method for estimating the mean of a Bernoulli random variable. We define the termination conditions for controlling the absolute and relative errors. We also briefly present a preliminary theoretical analysis of the proposed sampling method. Empirical simulation results show that our method often uses significantly lower sample size (i.e., the number of sampled instances) while maintaining competitive accuracy and confidence when compared with most existing methods such as that in [14]. Although the theoretical validity of the sampling method is only partially established. we strongly believe that our method should be sound in providing a rigorous guarantee that the estimation results under our scheme have desired accuracy and confidence.

Keywords: Sequential Sampling, Adaptive Sampling, Sample Size, Chernoff Bound, Hoeffding Bound.

1 Introduction

Random sampling is an important technique widely used in statistical analysis, computer science, machine learning and knowledge discovery. In statistics, random sampling is used to estimate the parameters of some underlying distribution by observing samples of certain size. In machine learning, researchers use sampling to estimate the accuracy of learned classifiers or to estimate features from vast amount of data. The problem of random sampling and parametric estimation is fundamental to statistics and relevant fields and has been studied extensively in the literature.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 220–229, 2011.

A key issue in designing a sampling scheme is to determine *sample size*, the number of sampled instances sufficient to assure the estimation accuracy and confidence. Well-known theoretical bounds such as the Chernoff-Hoeffding bound are commonly used in for computing the sufficient sample size. Conventional (batch) sampling methods are *static* in the sense that sufficient sample size is determined prior to the start of sampling. In contrast, adaptive sequential sampling does not fix the number of samples required in advance. A sequential sampling algorithm draws random samples one by one in an online sequential manner. It decides whether it has seen enough samples dependent on some measures related to the samples seen so far. This adaptive nature of sequential sampling method is attractive from both computational and practical perspectives. As shown in recent studies [2,4,14], adaptive sequential sampling could potentially reduce the sufficient sample sizes significantly compared to static batching sampling approaches. Clearly, it is desirable to keep the sample size small subject to the constraint of estimation accuracy and confidence. This would save not only computation time, but also the cost of generating the extra random samples when such costs are significant.

In this paper, we present a new adaptive sequential sampling method that often uses much lower sample size while maintaining competitive accuracy and confidence compared to most existing sampling methods.

Statisticians have investigated adaptive sampling procedures for estimating parameters under the heading of *sequential estimation*. However, existing sequential methods of parametric estimation in the statistics literature are dominantly of *asymptotic* nature, which may introduce unknown approximation errors to the necessary use of finite number of samples. Researchers in statistics and computer science have recently developed *adaptive sampling* schemes [7,8,14] that are of *non-asymptotic* nature for parametric estimation. Earlier works in Computer Science on adaptive sampling include the methods in [11,12,13] for estimating the size of a database query.

Efficient adaptive sampling has great potential applications to machine learning and data mining, especially when the underlying dataset is huge. For example, instead of using the entire huge data set for learning a target function, one can use sampling to get a subset of the data to construct a classifier. In Boosting, an ensemble learning method, the learning algorithm needs to select a "good" classifier with classification accuracy above $\frac{1}{2}$ at each boosting round. This would require estimating the accuracy of each classifier either exhaustively or by sampling. Watanabe et. al. have recently showed [7,8,14] successful application of their adaptive sampling methods to boosting.

In [2,4], an adaptive, multi-stage sampling framework has been proposed. The framework is more general than the adaptive sampling methods in [7,8,14]. Moreover, mathematical analysis and numerical computation indicate that the multistage estimation method in [2,4] improves upon the method in [14] by one to several orders of magnitude in terms of efficiency.

This current paper is closely related to the idea in [2,4]. The sampling schemes of [2,4] require computations (before sampling starts) to find the optimal value for the coverage tuning parameter ζ which is used in the stopping criterion for sampling. In contrast, the sequential sampling algorithms presented here require no computational effort to determine stopping rules. The benefits of the new approach is the simplicity of

implementation and the potential reduction in sample size as compared to other existing sampling methods such as that in [14].

The adaptive sequential sampling method presented here handles cases of controlling absolute error and relative error. Typically existing methods (including [14]) for handling absolute error recommend the use of *batch* sampling. However as we will show that adaptive sampling can also be used for absolute error control. We also show that our sampling method for controlling relative error has significantly lower sample size compared to that in [14].

2 Background

The basic problem tackled in this paper is to estimate the probability $p = \Pr(A)$ of a random event A from observational data. This is the same as estimating the mean $\mathbb{E}[X] = p$ of a Bernoulli random variable X in parametric estimation. We have access to i.i.d. samples $X_1, X_2, \cdots$ of the Bernoulli variable X such that $\Pr\{X = 1\} = 1 - \Pr\{X = 0\} = p$. An estimator for p can be taken as the *relative frequency* $\widehat{\boldsymbol{p}} = \frac{\sum_{i=1}^{\mathbf{n}} X_i}{\mathbf{n}}$, where $\mathbf{n}$ is the sample number at the termination of experiment. In the context of fixed-size sampling, the Chernoff-Hoeffding bound asserts that, for $\varepsilon, \delta \in (0, 1)$, the coverage probability $\Pr\{|\widehat{\boldsymbol{p}} - p| < \varepsilon\}$ is greater than $1 - \delta$ for any $p \in (0, 1)$ provided that $\mathbf{n} > \frac{\ln \frac{2}{\delta}}{2\varepsilon^2}$. Here ε is called the *margin of absolute error* and $1 - \delta$ is called the *confidence level*. Recently, an exact computational method has been established in [5] to substantially reduce this bound. To estimate p with a relative precision, it is a well-known result derived from the Chernoff bound that $\Pr\{|\widehat{\boldsymbol{p}} - p| < \varepsilon p\} > 1 - \delta$ provided that the pre-specified sample size $\mathbf{n}$ is greater than $\frac{3}{\varepsilon^2 p} \ln \frac{2}{\delta}$. Here $\varepsilon \in (0, 1)$ is called the *margin of relative error*. Since this sample size formula involves the value p which is exactly the one we wanted to estimate, its direct use is not convenient.

Chernoff-Hoeffding bounds have been used extensively in statistical sampling and Machine Learning. And in many cases, they are already quite tight bounds. However we are interested in doing even better than just using these bounds in the static way. We seek adaptive sampling schemes that allow us to achieve the goal of low sample size requirements without sacrifizing accuracy and confidence. In adaptive sampling, we draw some number of i.i.d. samples and test certain stopping criterion after seeing each new sample. The criterion for stopping sampling (and thus the bound on sufficient sample size) is determined with a formula dependent on the prescribed accuracy and confidence level, as well as the samples seen so far. In this paper, we shall consider the following two problems.

Problem 1 – Control of Absolute Error: Construct an adaptive sampling scheme such that, for *a priori* margin of absolute error $\varepsilon \in (0, 1)$ and confidence parameter $\delta \in (0, 1)$, the relative frequency $\widehat{\boldsymbol{p}}$ at the termination of the sampling process guarantees $\Pr\{|\widehat{\boldsymbol{p}} - p| < \varepsilon\} > 1 - \delta$ for any $p \in (0, 1)$.

Problem 2 – Control of Relative Error: Construct an adaptive sampling scheme such that, for *a priori* margin of relative error $\varepsilon \in (0, 1)$ and confidence parameter $\delta \in (0, 1)$, the relative frequency $\widehat{\boldsymbol{p}}$ at the termination of the sampling process guarantees $\Pr\{|\widehat{\boldsymbol{p}} - p| < \varepsilon p\} > 1 - \delta$ for any $p \in (0, 1)$.

The first problem has been treated in [2,3,4] as a special case of the general parameter estimation problem for a random variable X parameterized by $\theta \in \Theta$, aimed at obtaining estimator $\widehat{\boldsymbol{\theta}}$ for θ such that $\Pr\{|\widehat{\boldsymbol{\theta}} - \theta| < \varepsilon \mid \theta\} > 1 - \delta$ for any $\theta \in \Theta$. The approach proposed in [2,3,4] is outlined as follows. The sampling process consists of s stages with sample sizes $n_1 < n_2 < \cdots < n_s$. For $\ell = 1, \cdots, s$, define an estimator $\widehat{\boldsymbol{\theta}}_\ell$ for θ in terms of samples $X_1, \cdots, X_{n_\ell}$ of X. Let $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}_{\boldsymbol{l}}$, where $\boldsymbol{l}$ is the index of stage at the termination of sampling. In order to control the coverage probability by a positive parameter ζ, a general stopping rule has been proposed in [3, Theorem 2, Version 2, April 2, 2009] as "sampling is continued until $U_\ell - \varepsilon < \widehat{\boldsymbol{\theta}}_\ell < L_\ell + \varepsilon$" (or equivalently, $(L_\ell, U_\ell) \subseteq (\widehat{\boldsymbol{\theta}}_\ell - \varepsilon, \widehat{\boldsymbol{\theta}}_\ell + \varepsilon)$) for some $\ell \in \{1, \cdots, s\}$, where (L_ℓ, U_ℓ) is a confidence interval for θ with coverage probability greater than $1 - \zeta\delta$. This method of controlling the coverage probability was subsequently re-discovered by Jesse Frey in [9, 1st paragraph, Section 2] in the context of estimating the Bernoulli parameter p. Under the assumption that $\widehat{\boldsymbol{\theta}}$ is an estimator with a unimodal likelihood function, it has been established in [2, Theorem 8, Version 12, April 27, 2009] that the lower and upper bounds of $\Pr\{|\widehat{\boldsymbol{\theta}} - \theta| \geq \varepsilon \mid \theta\}$ for $\theta \in [a, b] \subseteq \Theta$ can be given, respectively, as $\Pr\{\widehat{\boldsymbol{\theta}} \leq a - \varepsilon \mid b\} + \Pr\{\widehat{\boldsymbol{\theta}} \geq b + \varepsilon \mid a\}$ and $\Pr\{\widehat{\boldsymbol{\theta}} \leq b - \varepsilon \mid a\} + \Pr\{\widehat{\boldsymbol{\theta}} \geq a + \varepsilon \mid b\}$. Adapted Branch & Bound Algorithm [2, Section 2.8, Version 12, April 27, 2009] and Adaptive Maximum Checking Algorithm [2, Section 3.3, Version 16, November 20, 2009] have been established in [2] for quick determination of whether $\Pr\{|\widehat{\boldsymbol{\theta}} - \theta| \geq \varepsilon \mid \theta\} \leq \delta$ for any $\theta \in \Theta$. A bisection search method was proposed in [2] to determine maximal ζ such that $\Pr\{|\widehat{\boldsymbol{\theta}} - \theta| < \varepsilon \mid \theta\} > 1 - \delta$ for any $\theta \in \Theta$. In a recent paper [9] devoted to the estimation of the Bernoulli parameter p, Jesse Frey has followed the general method of [2,3,4] for tuning the coverage probability. In particular, Jesse Frey re-discovered in [9, Appendix: The Checking Algorithm] exactly the Adaptive Maximum Checking Algorithm of [2] for determining whether the coverage probability associated with a given ζ is greater than $1 - \delta$ for $p \in (0, 1)$.

The second problem of estimating Bernoulli parameter p has been considered in [6] as a special case of estimating the mean of a bounded random variable. Let γ and ε be positive numbers. Let $X_i \in [0, 1],\ i = 1, 2, \cdots$ be i.i.d. random variables with common mean $\mu \in (0, 1)$. Let $\mathbf{n}$ be the smallest integer such that $\sum_{i=1}^{\mathbf{n}} X_i \geq \gamma$. It has been established in [6] that $\Pr\{|\frac{\gamma}{\mathbf{n}} - \mu| < \varepsilon\mu\} > 1 - \delta$ provided that $\gamma > \frac{(1+\varepsilon)\ln(2/\delta)}{(1+\varepsilon)\ln(1+\varepsilon)-\varepsilon}$.

For estimating p with margin of relative error $\varepsilon \in (0, 1)$ and confidence parameter $\delta \in (0, 1)$, Watanabe proposed in [14] to continue i.i.d. Bernoulli trials until A successes occur and then take the final relative frequency $\widehat{\boldsymbol{p}}$ as an estimator for p, where $A > \frac{3(1+\varepsilon)}{\varepsilon^2} \ln \frac{2}{\delta}$. We will show (empirically) in this paper that our proposed method for controlling relative error uses much smaller number of samples while maintaining competitive accuracy and confidence as compared to the adaptive sampling scheme of [14].

3 The Proposed Adaptive Sampling Method

In this section we describe a new, adaptive sampling method. Let us define the function $\mathscr{U}(z, \theta)$ which will be useful for defining our sampling scheme.

$$\mathscr{U}(z,\theta) = \begin{cases} z\ln\frac{\theta}{z} + (1-z)\ln\frac{1-\theta}{1-z} & z \in (0,\ 1),\ \ \theta \in (0,\ 1) \\ \ln(1-\theta) & z = 0,\ \theta \in (0,\ 1) \\ \ln(\theta) & z = 1,\ \theta \in (0,\ 1) \\ -\infty & z \in [0,\ 1],\ \theta \notin (0,\ 1) \end{cases}$$

Actually one can notice that the function $\mathscr{U}(z,\theta)$ equals $-DL(z||\theta)$ for $z, \theta \in (0,\ 1)$, where $DL(x||y) = x\ln\frac{x}{y} + (1-x)\ln\frac{1-x}{1-y}$ is the KL-divergence of two Bernoulli random variables with parameters x and y.

It is also not difficult to see that the function $\mathscr{U}(z,\ \theta)$ satisfies $\mathscr{U}(z,\ \theta) = \mathscr{U}(1-z,\ 1-\theta)$. It is known that $DL(p+x||p)$ is a convex function of x, and thus $\mathscr{U}(\theta+\varepsilon,\ \theta)$ is a *concave* function of ε. For simplicity of notations, define $w(x) = \frac{1}{2} - |\frac{1}{2} - x|$ for $x \in [0,\ 1]$.

3.1 Our Sampling Scheme for Controlling Absolute Error

Let $0 < \varepsilon < 1, 0 < \delta < 1$. The sampling scheme proceeds as follows.

Algorithm 1.
Let $\mathbf{n} \leftarrow 0,\ X \leftarrow 0$ and $\widehat{\boldsymbol{p}} \leftarrow 0$.
While $\mathbf{n} < \frac{\ln\frac{\delta}{2}}{\mathscr{U}(w(\widehat{\boldsymbol{p}}),\ w(\widehat{\boldsymbol{p}})+\varepsilon)}$
Do
begin
Draw a random sample Y according to the unknown distribution with parameter p.
Let $X \leftarrow X + Y,\ \mathbf{n} \leftarrow \mathbf{n} + 1$ and $\widehat{\boldsymbol{p}} \leftarrow \frac{X}{\mathbf{n}}$
end
Output $\widehat{\boldsymbol{p}}$ and $\mathbf{n}$.

It can be shown that the random sample size $\mathbf{n}$ at the termination of the sampling process must satisfy $n_1 = \left\lceil \frac{\ln(\delta)}{\ln(1-\varepsilon)} \right\rceil \le \mathbf{n} \le n_s = \left\lceil \frac{\ln\frac{1}{\delta}}{2\varepsilon^2} \right\rceil$ (as indicated in [2]). This algorithm can be viewed as an adaptation from the second sampling scheme proposed in Section 4.1.1 of [2, Version 20] by taking the coverage tuning parameter ζ as $\frac{1}{2}$ and the sample sizes at the various stages as consecutive integers between n_1 and n_s.

Properties of the sampling method - Conjecture 1
Based on a heuristic argument and extensive computational experiment, we believe that the following conjecture holds true: $\Pr\{|\widehat{\boldsymbol{p}} - p| < \varepsilon\} > 1 - \delta$, where $\widehat{\boldsymbol{p}}$ is the relative frequency when Algorithm 1 is terminated.

In the following Figures 1 and 2, using the recursive method proposed in [2,4], we have obtained the coverage probabilities and average sample numbers as functions of p for Algorithm 1 and the third stopping rule (with $\zeta = 2.1$ and consecutive sample sizes) proposed in Section 4.1.1 of [2] for $\varepsilon = 0.1,\ \delta = 0.05$. It can be seen from the green plot of Figure 1 that the above conjecture is indeed true for $\varepsilon = 0.1$ and $\delta = 0.05$. It is interesting to see that, for both stopping rules, the coverage probabilities are very close to the nominal level 0.95 for $p = 0.1$ and 0.9. However, for $0.1 < p < 0.9$, the coverage probabilities of Algorithm 1 is significantly greater than that of the stopping rule of [2]. As a consequence, the average sample number of Algorithm 1 is greater

than that of the stopping rule of [2]. This is clearly seen in Figure 2. It should be noted that the stopping rule of [2] is the most efficient stopping rule so far obtained by the computational method.

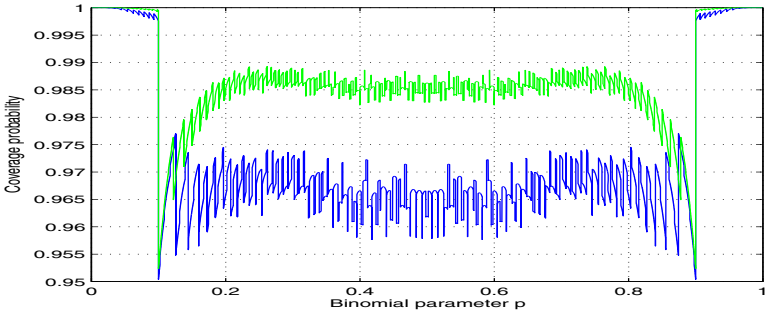


Fig. 1. Exact computation of coverage probability. The green plot is for Algorithm 1. The blue plot is for the stopping rule of [2].

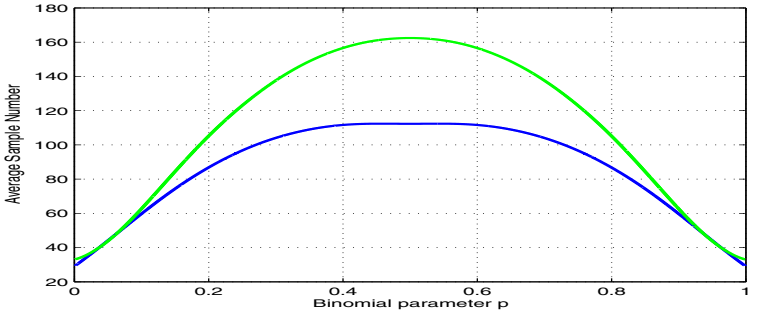


Fig. 2. Exact computation of average sample numbers. The green plot is for Algorithm 1. The blue plot is for the stopping rule of [2].

We have conducted a preliminary theoretical analysis on the properties of our sampling method and the following theorem summarizes the results for the case of absolute error. The proof of the theorem is skipped here due to lack of space.

Theorem 1. *Let* $n_0 = \max\{\lceil \frac{\ln \frac{\delta}{2}}{\mathscr{U}(p+\varepsilon,\ p+2\varepsilon)} \rceil,\ \lceil \frac{\ln \frac{\delta}{2}}{\mathscr{U}(p-\varepsilon, p-2\varepsilon)} \rceil\}$. *Assume that the true probability* p *to be estimated satisfies* $p \leq \frac{1}{2} - 2\varepsilon$. *Then with a probability of no less than* $1 - \frac{\delta}{2}$, *Algorithm 1 will stop with* $\mathbf{n} \leq n_0$ *samples and produce* $\widehat{\boldsymbol{p}}$ *which satisfies* $\widehat{\boldsymbol{p}} \leq p + \varepsilon$. *Similarly, if* $p \geq \frac{1}{2} + 2\varepsilon$, *with a probability no less than* $1 - \frac{\delta}{2}$, *the sampling algorithm will stop with* $\mathbf{n} \leq n_0$ *samples and produce* $\widehat{\boldsymbol{p}}$ *which satisfies* $\widehat{\boldsymbol{p}} \geq p - \varepsilon$.

To completely show the validity of the sampling method as outlined in Algorithm 1, we need to show that the sampling will not stop too early (stopping too early means

$\mathbf{n} \leq \frac{\ln \frac{\delta}{2}}{\mathscr{U}(p-\varepsilon,\, p)}$ when $p \leq \frac{1}{2}$ and $\mathbf{n} \leq \frac{\ln \frac{\delta}{2}}{\mathscr{U}(p+\varepsilon,\, p)}$ when $p > \frac{1}{2}$) to assure (with high confidence) good accuracy of the estimated parameter value. This turns out to be quite difficult to prove, although in our simulation experiments the chances of early stopping are apparently very small (much smaller than $\frac{\delta}{2}$).

However, we present some experimental results that show empirically our sampling method generates estimates with high accuracy and confidence. We used Matlab for the simulation experiment. In each experiment, a sequence of random samples is drawn according to the pre-determined probability p and Algorithm 1 is used to estimate $\widehat{\boldsymbol{p}}$ and decide (according to $\varepsilon,\ \delta$ and $\widehat{\boldsymbol{p}}$) whether to stop the experiment. The experiment is repeated 1000 times. Below we show histograms indicating the frequency of estimated values and the number of random samples used in the experiments. We used $\varepsilon = \delta = 0.1$ in simulations shown here.

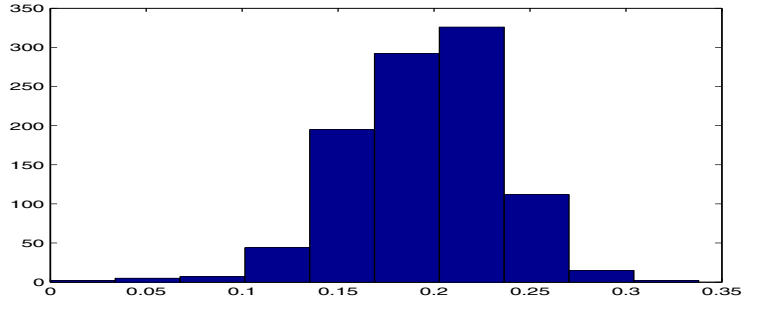


Fig. 3. Histogram for the estimated $\widehat{\boldsymbol{p}}$ values (p = 0.2, $\varepsilon = \delta = 0.1$). The vertical line shows the number of occurrences of the corresponding $\widehat{\boldsymbol{p}}$ values.

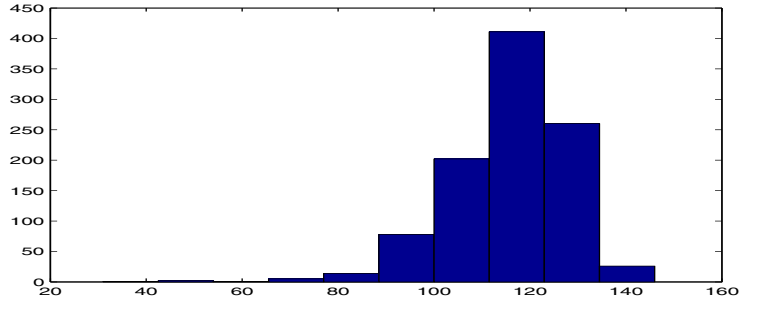


Fig. 4. Histogram for the random variable $\mathbf{n}$, the number of samples needed in each trial (p = 0.2, $\varepsilon = \delta = 0.1$)

To make sure that the results shown in these figures are not merely by luck, we show below a table illustrating the results of 9 simulations with $p = 0.1,\ 0.2,\ ...,\ 0.9$, $\varepsilon = \delta = 0.1$. Each simulation is a result of 1000 repeated experiments.

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\widehat{\Pr}\{\lvert\widehat{\boldsymbol{p}}-p\rvert \geq \varepsilon\}$	0	0.02	0.009	0.013	0.015	0.011	0.016	0.014	0
$\widehat{\boldsymbol{p}}$ max	0.19	0.312	0.447	0.553	0.644	0.731	0.838	1.000	1.000
$\widehat{\boldsymbol{p}}$ min	0.00	0.00	0.155	0.290	0.361	0.477	0.580	0.671	0.820
$\widehat{\boldsymbol{p}}$ mean	0.091	0.195	0.298	0.399	0.501	0.600	0.702	0.802	0.908
$\mathbf{n}$ max	115	141	150	150	150	150	150	143	111
$\mathbf{n}$ min	30	30	103	138	146	134	103	30	30
$\mathbf{n}$ mean	77	115	138	148	149	148	138	115	78

From the experimental data, it seems that the expected value for the random variable $\mathbf{n}$ is around $\frac{\ln\frac{\delta}{2}}{\mathscr{U}(p,\, p+\varepsilon)}$ for $p \leq \frac{1}{2}$, and it is around $\frac{\ln\frac{\delta}{2}}{\mathscr{U}(p,\, p-\varepsilon)}$ for $p > \frac{1}{2}$. This is reasonable if the $\widehat{\boldsymbol{p}}$ generated by the sampling algorithm is around the true probability p. And indeed we can see from the above table that the mean value of $\widehat{\boldsymbol{p}}$ is very close to the true parameter value p.

3.2 Our Sampling Scheme for Controlling Relative Error

Given $0 < \varepsilon < 1$, $0 < \delta < 1$, the sampling proceeds as follows.

Algorithm 2.
Let $\mathbf{n} \leftarrow 0,\ X \leftarrow 0$ and $\widehat{\boldsymbol{p}} \leftarrow 0$.
While $\widehat{\boldsymbol{p}} = 0$ *or* $\mathbf{n} < \frac{\ln\frac{\delta}{2}}{\mathscr{U}(\widehat{\boldsymbol{p}},\, \frac{\widehat{\boldsymbol{p}}}{1+\varepsilon})}$
Do
begin
Draw a random sample Y according to the unknown distribution with parameter p.
Let $X \leftarrow X + Y,\ \mathbf{n} \leftarrow \mathbf{n} + 1$ and $\widehat{\boldsymbol{p}} \leftarrow \frac{X}{\mathbf{n}}$
end
Output $\widehat{\boldsymbol{p}}$ and $\mathbf{n}$.

Actually, Algorithm 2 was inspired by the multistage sampling scheme proposed in [2, Theorem 23, Version 20], which can be described as "continue sampling until $\mathscr{U}(\widehat{\boldsymbol{p}}_\ell,\, \frac{\widehat{\boldsymbol{p}}_\ell}{1+\varepsilon}) \leq \frac{\ln(\zeta\delta_\ell)}{n_\ell}$ at some stage with index ℓ and then take the final relative frequency as the estimator for p", where n_ℓ and $\widehat{\boldsymbol{p}}_\ell$ are respectively the sample size and the relative frequency at the ℓ-th stage, and δ_ℓ is dependent on ℓ such that $\delta_\ell = \delta$ for ℓ no greater than some pre-specified integer τ and that $\delta_\ell = \delta 2^{\tau-\ell}$ for ℓ greater than τ.

Properties of Algorithm 2 - Conjecture 2
Based on a heuristic argument and extensive computational experiment, we believe that the following conjecture holds true: $\Pr\{|\widehat{\boldsymbol{p}} - p| < \varepsilon p\} > 1 - \delta$, where $\widehat{\boldsymbol{p}}$ is the relative frequency when the sampling of Algorithm 2 is terminated.

Let $N_1 = \max\{\lceil\frac{\ln\frac{\delta}{2}}{\mathscr{U}(p(1-\varepsilon),\, p)}\rceil,\ \lceil\frac{\ln\frac{\delta}{2}}{\mathscr{U}(p(1+\varepsilon),\, p)}\rceil\}$ and $N_2 = \lceil\frac{\ln\frac{\delta}{2}}{\mathscr{U}(p(1-\varepsilon),\, \frac{p(1-\varepsilon)}{1+\varepsilon})}\rceil$. One can show that if Algorithm 2 terminates with $N_1 \leq \mathbf{n} \leq N_2$, then $|\frac{\widehat{\boldsymbol{p}}-p}{p}| \leq \varepsilon$. As of now we have the following result:

Theorem 2. *With a probability no less than* $1 - \frac{\delta}{2}$*, Algorithm 2 will stop with* $n \leq N_2$ *and produce* $\widehat{\boldsymbol{p}} \geq p(1-\varepsilon)$.

We would desire to show that the Algorithm will stop between N_1 and N_2 steps with high probability. What remains to be proven is that with high probability, the algorithm will NOT stop too early and produce an estimate $\widehat{\boldsymbol{p}}$ which is bigger than $p(1+\varepsilon)$. Similar to the absolute-error case, this is not so easy to prove. However, we will show empirical results to support the conjecture that the algorithm indeed will stop after N_1 steps (most of the time). Moreover we will show simulation results comparing our method and the adaptive sampling method in [14].

In the following table we show the simulation results using $p = 0.1, 0.2, ..., 0.7$ and we fixed $\varepsilon = 0.2$ and $\delta = 0.1$ in these simulations. The columns labeled as "CC" are results of using Algorithm 2, whereas the columns labeled as "Wata" are results of the method of Watanabe in [14]. As in the absolute error case, each simulation is a result of 1000 repeated experiments.

p	0.1		0.2		0.3		0.4		0.5		0.6		0.7	
	CC	Wata	CC	Wata	CC	Wata	CC	Wata	CC	Wata	CC	Wata	CC	Wata
$\widehat{\Pr}$	0.014	0.001	0.006	0	0.004	0.001	0.006	0	0.009	0	0.010	0	0.014	0
$\widehat{\boldsymbol{p}}$ max	0.129	0.119	0.250	0.237	0.385	0.363	0.515	0.465	0.664	0.595	0.822	0.676	0.968	0.792
$\widehat{\boldsymbol{p}}$ min	0.075	0.075	0.161	0.168	0.248	0.248	0.321	0.345	0.407	0.429	0.498	0.528	0.558	0.620
$\widehat{\boldsymbol{p}}$ mean	0.101	0.101	0.201	0.200	0.302	0.299	0.401	0.399	0.503	0.499	0.604	0.600	0.708	0.699
$\mathbf{n}$ max	2248	2248	1017	1125	605	750	427	563	302	451	215	376	174	522
$\mathbf{n}$ min	1321	1897	595	850	327	620	204	484	119	378	62	333	31	284
$\mathbf{n}$ mean	1743	2189	788	1099	470	735	311	552	213	442	150	369	102	317

4 Conclusions and Future Work

In this paper we present a new, adaptive sampling technique for estimating the mean of a random Bernoulli variable. We define termination conditions for controlling the absolute and relative errors. Preliminary theoretical analysis results about the proposed sampling method are also outlined. We also present empirical simulation results indicating that our method often uses significantly lower sample size while maintaining competitive estimation accuracy and confidence compared with most existing approaches such as that of [14].

The theoretical analysis presented in this paper show that with high probability the new sampling method will stop without using excessive number of samples and producing an estimate with low one-sided error. What remains to be proven is that the proposed method will, with high probability, stop with sufficient number of samples so that the estimate produced will have low error on two-sides. This conjecture appears to be supported by the empirical studies, but a theoretical result will be better. We strongly believe that our method should be sound in providing a rigorous guarantee that the estimation results under our scheme have desired accuracy and confidence. Another direction to pursue is to explore the use of the new sampling method in various machine learning tasks. For example, it is worthwhile to see the application of our sampling method to the boosting problem as done in [7].

References

1. Chernoff, H.: A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. Ann. Math. Statist. 23, 493–507 (1952)
2. Chen, X.: A new framework of multistage estimation, arXiv.0809.1241 (math.ST)
3. Chen, X.: Confidence interval for the mean of a bounded random variable and its applications in point estimation, arXiv:0802.3458 (math.ST)
4. Chen, X.: A new framework of multistage parametric inference. In: Proceeding of SPIE Conference, Orlando, Florida, vol. 7666, pp. 76660R1–12 (April 2010)
5. Chen, X.: Exact computation of minimum sample size for estimation of binomial parameters. Journal of Statistical Planning and Inference 141, 2622–2632 (2011), http://arxiv.org/abs/0707.2113
6. Chen, X.: Inverse sampling for nonasymptotic sequential estimation of bounded variable means, arXiv:0711.2801 (math.ST) (November 2007)
7. Domingo, C., Watanabe, O.: Scaling up a boosting-based learner via adaptive sampling. In: Knowledge Discovery and Data Mining, pp. 317–328. Springer, Heidelberg (2000)
8. Domingo, C., Watanabe, O.: Adaptive sampling methods for scaling up knowledge discovery algorithms. In: Proceedings of 2nd Int. Conference on Discovery Science, Japan (December 1999)
9. Frey, J.: Fixed-width sequential confidence intervals for a proportion. The American Statistician 64, 242–249 (2010)
10. Hoeffding, W.: Probability inequalities for sums of bounded variables. J. Amer. Statist. Assoc. 58, 13–29 (1963)
11. Lipton, R., Naughton, J., Schneider, D.A., Seshadri, S.: Efficient sampling strategies for relational database operations. Theoretical Computer Science 116, 195–226 (1993)
12. Lipton, R., Naughton, J.: Query size estimation by adaptive sampling. Journal of Computer and System Sciences 51, 18–25 (1995)
13. Lynch, J.F.: Analysis and applicaion of adaptive sampling. Journal of Computer and System Sciences 66, 2–19 (2003)
14. Watanabe, O.: Sequential sampling techniques for algorithmic learning theory. Theoretical Computer Science 348, 3–14 (2005)

An Evolutionary Algorithm for Global Induction of Regression Trees with Multivariate Linear Models

Marcin Czajkowski and Marek Kretowski

Faculty of Computer Science, Bialystok University of Technology
Wiejska 45a, 15-351 Białystok, Poland
{m.czajkowski,m.kretowski}@pb.edu.pl

Abstract. In the paper we present a new evolutionary algorithm for induction of regression trees. In contrast to the typical top-down approaches it globally searches for the best tree structure, tests at internal nodes and models at the leaves. The general structure of proposed solution follows a framework of evolutionary algorithms with an unstructured population and a generational selection. Specialized genetic operators efficiently evolve regression trees with multivariate linear models. Bayesian information criterion as a fitness function mitigate the over-fitting problem. The preliminary experimental validation is promising as the resulting trees are less complex with at least comparable performance to the classical top-down counterpart.

Keywords: model trees, evolutionary algorithms, multivariate linear regression, BIC.

1 Introduction

The most common predictive tasks in data mining applications are classification and regression [5]. One of the most widely used prediction techniques are decision trees [19]. Regression and model trees may be considered as a variant of decision trees, designed to approximate real-valued functions instead of being used for classification tasks. Main difference between a typical regression tree and a model tree is that, for the latter, terminal node is replaced by a regression plane instead of a constant value. Those tree-based approaches are now popular alternatives to classical statistical techniques like standard regression or logistic regression.

In this paper we want to investigate a global approach to model tree induction based on a specialized evolutionary algorithm. This solution extends our previous research on evolutionary classification and regression trees which showed that a global induction could be more adequate in certain situations. Our work covers the induction of univariate trees with multivariate linear models at the leaves.

1.1 Global Versus Local Induction

Linear regression is a global model in which the single predictive function holds over the entire data-space [9]. However many regression problems cannot be

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 230–239, 2011.

solved by a single regression model especially when the data has many attributes which interact in a complicated ways. Recursively partitioning the data and fitting local models to the smaller regions, where the interactions are more simple, is a good alternative to complicated, nonlinear regression approaches. The recursive partitioning [16] may be realized by top-down induced regression trees. Starting from the root node they search for the locally optimal split (test) according to the given optimality measure and then the training data is redirected to newly created nodes. This procedure is recursively repeated until the stopping criteria are met and in each of the terminal node called leaf, a locally optimal model is built for each region. Finally, the post-pruning is applied to improve the generalization power of the predictive model. Such a technique is fast and generally efficient in many practical problem, but obviously does not guarantee the globally optimal solution. Due to the greedy nature, algorithms may not generate the smallest possible number of rules for a given problem [17] and a large number of rules results in decreased comprehensibility. Therefore, in certain situations more global approach could lead to improvement in prediction and size of the resulting models.

1.2 Related Work

The *CART* system [2] is one of most known top-down induced prediction tree. The *CART* algorithm finds a split that minimizes the Residual Sum of Squares (RSS) of the model when predicting. Next, it builds a piecewise constant model with each terminal node fitted by the training sample mean. The *CART* algorithm was later improved by replacing single predicted values in the leaves by more advanced models like in *SECRET* [4] or *RT* [21]. The most popular system which induce top-down model tree is *M5* [23]. Like *CART*, it builds tree-based models but, whereas regression trees have values at their leaves, the tree constructed by *M5* can have multivariate linear models analogous to piecewise linear functions.

One of the first attempts to optimize the overall RSS was presented in *RETRIS* [8] model tree. Algorithm simultaneously optimized the split and the models at the terminal nodes to minimize the global RSS. However *RETRIS* is not scalable and does not support larger datasets because of the huge complexity [17]. More recent solution called *SMOTI* [14] allows regression models to exist not only in leaves but also in the upper parts of the tree. Authors claim that this allows for individual predictors to have both global and local effects on the model tree.

Our previously performed research showed that evolutionary inducers are capable to efficiently induce various types of classification trees: univariate [10], oblique [11] and mixed [12]. In our last papers we applied a similar approach to obtain accurate and compact regression trees [13] and we did preliminary experiments with the model trees that have simple linear regression models at the leaves [3].

Proposed solution denoted as *GMT* improved our previous work: starting with more heterogenous population, additional genetic operators and a new fitness

function based on Bayesian information criterion (*BIC*) [20]. The models at the leaves were extended from simple linear to multivariate linear regression models.

2 An Evolutionary Induction of Model Trees

Structure of the proposed solution follows a typical framework of evolutionary algorithms [15] with an unstructured population and a generational selection.

2.1 Representation

Model trees are represented in their actual form as typical univariate trees, similarly as in our previous work [3]. Each test in a non-terminal node concerns only one attribute (nominal or continuous valued). In case of a continuous-valued feature typical inequality tests are applied. As for potential splits only the precalculated candidate thresholds are considered. A candidate threshold for the given attribute is defined as a midpoint between such a successive pair of examples in the sequence sorted by the increasing value of the attribute, in which the examples are characterized by different predicted values. Such a solution significantly limits the number of possible splits. For a nominal attribute at least one value is associated with each branch. It means that an inner disjunction is built into the induction algorithm.

At each leaf a multivariate linear model is constructed using standard regression technique [18] with cases and feature vectors associated with that node. A dependent variable y is now explained not by single variable like in [3] but a linear combination of multiple independent variables $x_1, x_2, \ldots, x_p$:

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \ldots + \beta_p * x_p \tag{1}$$

where p is the number of independent variables, x_i are independent variables, β_i are fixed coefficients that minimizes the sum of squared residuals of the model.

Additionally, in every node information about learning vectors associated with the node is stored. This enables the algorithm to perform more efficiently local structure and tests modifications during applications of genetic operators.

2.2 Initialization

The initial tree construction is similar to the typical approaches like *CART* and *M5*. At first, we construct a standard regression tree with local means of dependent variable values from training objects in every leaf. Initial individuals are created by applying the classical top-down algorithm. The recursive partitioning is finished when all training objects in node are characterized by the same predicted value (or it varies only slightly [23]) or the number of objects in a node is lower than the predefined value (default value: 5). Additionally, user can set the maximum tree depth (default value: 10). Next, a multivariate linear model is built at each terminal node.

An appropriate trade off between a degree of heterogeneity and a computation time is obtained by various data-driven manners for selecting attributes and choosing search strategies in non-terminal nodes:

- initial individuals are induced from randomly chosen subsamples of the original training data (10% of data, but not more than 500 examples);
- each individual searches splitting tests from randomly chosen subsamples of the attribute set (50% of attributes);
- one of three test search strategies in non-terminal nodes is applied [3]:
  - *Least Squares (LS)* reduction,
  - *Least Absolute Deviation (LAD)* reduction,
  - *dipolar*, where a dipol (a pair of feature vectors) is selected and then a test is constructed which splits this dipole. Selection of the dipole is randomized but longer (with bigger difference between dependent variable values) dipoles are preferred and mechanism similar to the ranking linear selection [15] is applied.

2.3 Genetic Operators

Like in our previous papers [3] [13] we have applied two specialized genetic operators corresponding to the classical mutation and cross-over. Both operators affect the tree structure, tests in non-terminal nodes and models at leaves. After each evolutionary iteration it is usually necessary to relocate learning vectors between parts of the tree rooted in the altered node. This can cause that certain parts of the tree does not contain any learning vectors and has to be pruned.

Cross-over. Cross-over solution starts with selecting positions in two affected individuals. In each of two trees one node is chosen randomly. We have proposed three variants of recombination:

- subtrees starting in the selected nodes are exchanged,
- tests associated with the nodes are exchanged (only when non-terminal nodes are chosen and the number of outcomes are equal),
- branches which start from the selected nodes are exchanged in random order (only when non-terminal nodes are chosen and the number of outcomes are equal).

Mutation. Mutation solution starts with randomly choosing the type of node (equal probability to select leaf or internal node). Next, the ranked list of nodes of the selected type is created and a mechanism analogous to ranking linear selection [15] is applied to decide which node will be affected. Depending on the type of node, ranking take into account:

- absolute error - worse in terms of accuracy leaves and internal nodes are mutated with higher probability (homogenous leaves are not included),

– location (level) of the internal node in the tree - it is evident that modification of the test in the root node affects whole tree and has a great impact, whereas mutation of an internal node in lower parts of the tree has only a local impact. Therefore, internal nodes in lower parts of the tree are mutated with higher probability.

We have proposed new mutation operators for internal node:

– tests between father and son exchanged,
– symmetric mutation between sub-trees,
– test in node changed by: new random one or new dipolar (described in section 2.2),
– shifting the splitting threshold (continuous-valued feature) or re-grouping feature values (nominal features),
– node can be transformed (pruned) into a leaf,

and for the leaves:

– transform leaf into an internal node with a new dipolar test,
– extend linear model by adding new randomly chosen attribute,
– simplify linear model by removing the randomly chosen attribute,
– change linear model attributes with random ones,
– delete from linear model the least important attribute.

After performed mutation in internal nodes the models in corresponding leaves are not recalculated because adequate linear models can be found while performing the mutations at the leaves. Modifying and recalculating leaf model makes sense only if it contains objects with different dependent variable values or different independent variables that build the linear model.

2.4 Selection and Termination Condition

Evolution terminates when the fitness of the best individual in the population does not improve during the fixed number of generations. In case of a slow convergence, maximum number of generations is also specified, which allows us to limit computation time.

Ranking linear selection [15] is applied as a selection mechanism. Additionally, in each iteration, single individual with the highest value of fitness function in current population in copied to the next one *(elitist strategy)*.

2.5 Fitness Function

A fitness function drives evolutionary search process and therefore is one of the most important and sensitive component of the algorithm. Direct minimization of the prediction error measured on the learning set usually leads to the overfitting problem. In a typical top-down induction of decision trees, this problem is partially mitigated by defining a stopping condition and by applying a post-pruning.

In our previous works [3] we used Akaike's information criterion (AIC) [1] as the fitness function. Performed experiments suggested that penalty for increasing model size should depend on the number of observations in the data. Therefore we have replaced AIC with the Bayesian information criterion (BIC) [20] that is given by:

$$Fit_{BIC}(T) = -2 * ln(L(T)) + ln(n) * k(T) \tag{2}$$

where $L(T)$ is the maximum of the likelihood function of the tree T, $k(T)$ is the number of model parameters and n is the number of observations. The log(likelihood) function $L(T)$ is typical for regression models [7] and can be expressed as:

$$ln(L(T)) = -0.5n * [ln(2\pi) + ln(SS_e(T)/n) + 1] \tag{3}$$

where $SS_e(T)$ is the sum of squared residuals of the tree T. The term $2 * k(T)$ can also be viewed as a penalty for over-parametrization and has to include not only the tree size but also the number of attributes that build models at the leaves. The number of independent parameters $k(T)$ in the complexity penalty term is equal $2 * (Q(T) + M(T))$ where $Q(T)$ is the number of nodes in model tree T and $M(T)$ is the sum of all attributes in the linear models at the leaves.

3 Experimental Validation

In this section, we study the predictive accuracy and size of the proposed approach (denoted as *GMT*) to other methods. Validation was performed on synthetical and real-life datasets. Since our algorithm induces model trees we have compared it against the popular *M5* [23] counterpart. The *M5* algorithm has the same tree structure: univariate splits and multivariate linear models at the leaves, as the *GMT*. The most important difference between both solution is the tree construction where the *M5* is a traditional greedy top-down inducer and the *GMT* approach searches for optimal trees in a global manner by using an evolutionary algorithm. We also included results obtained by the *REPTree* which is another classical top-down inducer. *REPTree* builds a regression tree using variance and prunes it using reduced-error pruning (with backfitting). Both comparative algorithms are run using the implementations in WEKA [6], software that is publicly available.

Each tested algorithm run with default values of parameters through all datasets. All presented results correspond to averages of 20 runs and were obtained by using test sets (when available) or by 10-fold cross-validation. Root mean squared error (RMSE) is given as the error measure of the algorithms. The number of nodes is given as a complexity measure (size) of regression and model trees.

3.1 Synthetical Datasets

In the first group of experiments, two simple artificially generated datasets with analytically defined decision borders are analyzed. Both datasets contain a

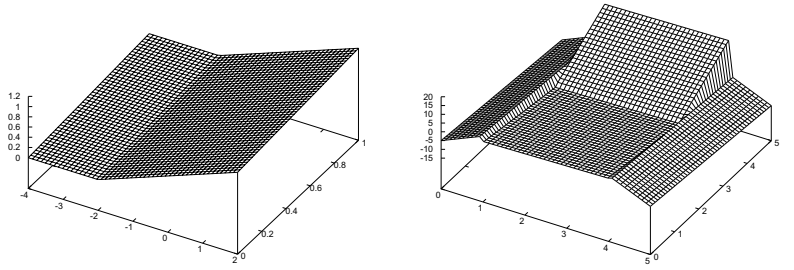


Fig. 1. Examples of artificial datasets (*split plane* - left, *armchair3* - right)

feature that is linearly dependent with one of two independent features. One thousand observations for each dataset were divided into a training set (33.3% of observations) and testing set (66.7%).

The artificial dataset *split plane* that is illustrated in the Fig. 1 can be perfectly predictable with regression lines on subsets of the data resulting from a single partition. The equation is:

$$y(x_1, x_2) = \begin{cases} 0.2 * x_2 & x_1 < -2 \\ 0.25 * x_1 + 0.2 * x_2 + 0.5 & x_1 \geq -2 \end{cases} \quad (4)$$

The test in the root node for both greedy top-down inducers is not optimal. *M5* approach minimizes the combined standard deviation of both partitions of each subset and sets first split at threshold $x_1 = -1.18$. *REPTree* is using the CART approach, partitions this dataset at $x_1 = -0.44$ minimizing the RSS and has size equal 88. *GMT* partitions the data at threshold $x_1 = -2.00$ because it is able to search globally for the best solution. This simple artificial problem illustrates general advantage of the global search solution to greedy algorithms. The induced *GMT* and *M5* trees are illustrated in Figure 2 and the Table 1 presents the generated multivariate linear models at the leaves.

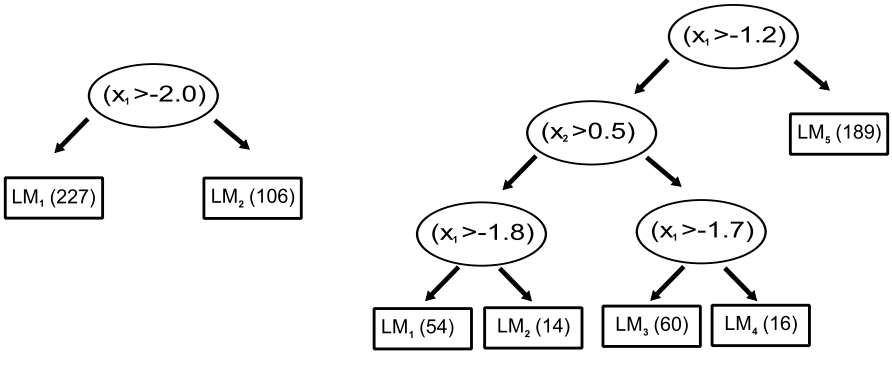


Fig. 2. Examples of model trees for *split plane* (GMT - left, M5 - right)

Table 1. Generated multivariate linear models for *GMT* and *M5*

GMT	LM_1: $y(x_1, x_2) = 0.25 * x_1 + 0.2 * x_2 + 0.5$
	LM_2: $y(x_1, x_2) = 0.2 * x_2 + 0.5$
M5	LM_1: $y(x_1, x_2) = 0.1865 * x_2 + 0.0052$
	LM_2: $y(x_1, x_2) = 0.25 * x_1 + 0.2 * x_2 + 0.5$
	LM_3: $y(x_1, x_2) = 0.1936 * x_2 + 0.0079$
	LM_4: $y(x_1, x_2) = 0.25 * x_1 + 0.2 * x_2 + 0.5$
	LM_5: $y(x_1, x_2) = 0.25 * x_1 + 0.2 * x_2 + 0.5$

Illustrated in the Fig. 1 dataset *Armchair3* is more complex than *split plane*. Many traditional approaches will fail to efficiently split the data as the greedy inducers search only for a locally optimal solutions. The equation is:

$$y(x_1, x_2) = \begin{cases} 10 * x_1 - 1.5 * x_2 - 5 & x_1 < 1 \\ -10 * x_1 - 1.5 * x_2 + 45 & x_1 \geq 4 \\ 0.5 * x_1 - 2.5 * x_2 + 1.5 & x_2 < 3;\ 1 \leq x_1 < 4 \\ 0.5 * x_1 + 10 * x_2 - 35 & x_2 \geq 3;\ 1 \leq x_1 < 4 \end{cases} \tag{5}$$

Similarly to previous experiment, *GMT* managed to find the best split at $x_1 = 1.00$ and induced optimal model tree. *M5* to build the tree needed 18 rules at the leaves and the first split threshold was set at $x_1 = 3.73$. *REPTree* using the CART approach has the first data partition at threshold $x_1 = 4.42$ and has a tree size equal 87.

3.2 Real-Life Datasets

Second group of experiments include several real-life datasets from UCI Machine Learning Repository [22]. Application of the *GMT* to the larger datasets showed that in contrast to RETRIS [8] our method scales well. In the proposed solution the smoothing function is not yet introduced therefore for more honest comparison we present the results of the unsmoothed *M5* and smoothed *M5 smot.* algorithm. The *REPTree* which is another classical top-down inducer build only regression trees and therefore has lower predictive accuracy. Table 2 presents characteristics of investigated datasets and obtained results.

It can be observed that on the real-life datasets the *GMT* managed to induce significantly smaller trees, similarly to the results on artificial data. Additionally, even without smoothing process that improves the prediction accuracy of tree-based models [23], *GMT* has at least comparable performance to smoothed *M5* and on two out of six datasets (*Elevators* and *Kinemaics*) is significantly better. The percentage deviation of $RMSE$ for *GMT* on all datasets was under 0.5%. As for the *REPTree* we may observe that the regression tree is no match for both model trees. Higher model comprehensibility of the *REPTree* thanks to simplified models at leaves is also doubtful because of the large tree size.

As with evolutionary data-mining systems, the proposed approach is more time consuming than the classical top-down inducers. However, experiments performed with typical desktop machine (Dual-Core CPU 1.66GHz with 2GB RAM)

Table 2. Characteristics of the real-life datasets (number of objects/number of numeric features/number of nominal features) and obtained results

		GMT		*M5*		*M5 smot.*		*REPTree*	
Dataset	Properties	RMSE	size	RMSE	size	RMSE	size	RMSE	size
Abalone	4177/7/1	2.150	**3.8**	2.134	12	**2.130**	12	2.358	201
Ailerons	13750/40/0	0.000165	**4.2**	**0.000164**	5.0	**0.000164**	5.0	0.000203	553
Delta Ailerons	7129/5/0	**0.000164**	**9.5**	0.000167	22	0.000165	22	0.000175	291
Delta Elevators	9517/6/0	**0.001424**	**3.1**	0.001427	8.0	0.001426	8.0	0.00150	319
Elevators	16599/18/0	**0.002448**	**14**	0.002702	45	0.002670	45	0.003984	503
Kinemaics	8192/8/0	**0.1457**	**24**	0.1654	106	0.1600	106	0.1906	819

showed that the calculation time even for the largest datasets are acceptable (from 5 minutes for the *Abalone* to around 2 hours for the *Ailerons*).

4 Conclusion

This paper presents a new global approach to the model tree learning. In contrast to classical top-down inducers, where locally optimal tests are sequentially chosen, in *GMT* the tree structure, tests in internal nodes and models at the leaves are searched in the same time by specialized evolutionary algorithm. This way the inducer is able to avoid local optima and to generate better predictive model. Even preliminary experimental results show that the globally evolved regression models are competitive compared to the top-down based counterparts, especially in the term of tree size.

Proposed approach is constantly improved. Further research to determine more appropriate value of complexity penalty term in the *BIC* criterion is advised and other commonly used measures should be considered. Currently we are working on a smoothing process that will improve prediction accuracy. On the other hand, we plan to introduce oblique tests in the non-terminal nodes and more advance models at the leaves.

Acknowledgments. This work was supported by the grant S/WI/2/08 from Bialystok University of Technology.

References

1. Akaike, H.: A New Look at Statistical Model Identification. IEEE Transactions on Automatic Control 19, 716–723 (1974)
2. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth Int. Group, Belmont (1984)
3. Czajkowski, M., Kretowski, M.: Globally Induced Model Trees: An Evolutionary Approach. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 324–333. Springer, Heidelberg (2010)

4. Dobra, A., Gehrke, J.: SECRET: A Scalable Linear Regression Tree Algorithm. In: Proc. of KDD 2002 (2002)
5. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park (1996)
6. Frank, E., et al.: Weka 3 - Data Mining with Open Source Machine Learning Software in Java. University of Waikato (2000), http://www.cs.waikato.ac.nz/~ml/weka
7. Gagne, P., Dayton, C.M.: Best Regression Model Using Information Criteria. Journal of Modern Applied Statistical Methods 1, 479–488 (2002)
8. Karalic, A.: Linear Regression in Regression Tree Leaves. International School for Synthesis of Expert Knowledge, Bled, Slovenia (1992)
9. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Data Mining, Inference and Prediction, 2nd edn. Springer, Heidelberg (2009)
10. Kretowski, M., Grześ, M.: Global Learning of Decision Trees by an Evolutionary Algorithm. Information Processing and Security Systems, 401–410 (2005)
11. Kretowski, M., Grześ, M.: Evolutionary Learning of Linear Trees with Embedded Feature Selection. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 400–409. Springer, Heidelberg (2006)
12. Kretowski, M., Grześ, M.: Evolutionary Induction of Mixed Decision Trees. International Journal of Data Warehousing and Mining 3(4), 68–82 (2007)
13. Kretowski, M., Czajkowski, M.: An Evolutionary Algorithm for Global Induction of Regression Trees. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS, vol. 6114, pp. 157–164. Springer, Heidelberg (2010)
14. Malerba, D., Esposito, F., Ceci, M., Appice, A.: Top-down Induction of Model Trees with Regression and Splitting Nodes. IEEE Transactions on PAMI 26(5), 612–625 (2004)
15. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer, Heidelberg (1996)
16. Murthy, S.: Automatic construction of decision trees from data: A multi-disciplinary survey. Data Mining and Knowledge Discovery 2, 345–389 (1998)
17. Potts, D., Sammut, C.: Incremental Learning of Linear Model Trees. Machine Learning 62, 5–48 (2005)
18. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C. Cambridge University Press, Cambridge (1988)
19. Rokach, L., Maimon, O.Z.: Data mining with decision trees: theory and application. Machine Perception Arfitical Intelligence 69 (2008)
20. Schwarz, G.: Estimating the Dimension of a Model. The Annals of Statistics 6, 461–464 (1978)
21. Torgo, L.: Inductive Learning of Tree-based Regression Models. Ph.D. Thesis, University of Porto (1999)
22. Blake, C., Keogh, E., Merz, C.: UCI Repository of Machine Learning Databases (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
23. Quinlan, J.: Learning with Continuous Classes. In: Proc. of AI 1992, pp. 343–348. World Scientific, Singapore (1992)

Optimizing Probabilistic Models for Relational Sequence Learning

Nicola Di Mauro, Teresa M.A. Basile, Stefano Ferilli, and Floriana Esposito

Department of Computer Science, LACAM laboratory
University of Bari "Aldo Moro", Via Orabona,4, 70125 Bari, Italy
{ndm,basile,ferilli,esposito}@di.uniba.it

Abstract. This paper tackles the problem of relational sequence learning selecting relevant features elicited from a set of labelled sequences. Each relational sequence is firstly mapped into a feature vector using the result of a feature construction method. The second step finds an optimal subset of the constructed features that leads to high classification accuracy, by adopting a wrapper approach that uses a stochastic local search algorithm embedding a Bayes classifier. The performance of the proposed method on a real-world dataset shows an improvement compared to other sequential statistical relational methods, such as Logical Hidden Markov Models and relational Conditional Random Fields.

1 Introduction

Sequential data may be found in many contexts of everyday life, and in many computer science applications such as video understanding, planning, computational biology, user modelling and speech recognition. Different methodologies have been proposed to face the problem of sequential learning. Some environments involve very complex components and features, and hence classical existing approaches have been extended to the case of relational sequences [1] to exploit a more powerful representation formalism. Sequential learning techniques may be classified according to the language they adopt to describe sequences. On the one hand there are methods adopting a propositional language, such as Hidden Markov Models (HMMs), allowing both a simple model representation and an efficient algorithm; on the other hand (Sequential) Statistical Relational Learning (SRL) [2] techniques, such as Logical Hidden Markov Models (LoHMMs) [3] and relational Conditional Random Fields [4, 5] are able to elegantly handle complex and structured descriptions for which a flat representation could make the problem intractable to propositional techniques. The goal of this paper is to propose a new probabilistic method for relational sequence learning [1].

A way to tackle the task of inferring discriminant functions in relational learning is to reformulate the problem into an attribute-value form and then apply a propositional learner [6]. The reformulation process may be obtained adopting a *feature construction* method, such as mining frequent patterns that can then be successfully used as new Boolean features [7–9]. Since, the effectiveness of learning algorithms strongly depends on the used features, a *feature selection*

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 240–249, 2011.

task is very desirable. The aim of feature selection is to find an optimal subset of the input features leading to high classification performance, or, more generally, to carry out the classification task optimally. However, the search for a variable subset is a NP-hard problem. Therefore, the optimal solution cannot be guaranteed to be reached except when performing an exhaustive search in the solution space. Using *stochastic local search* procedures [10] allows one to obtain good solutions without having to explore the whole solution space.

In this paper we propose an algorithm for relational sequence learning, named `Lynx`[1], that works in two phases. In the first phase it adopts a feature construction approach that provides a set of probabilistic features. In the second step, `Lynx` adopts a wrapper feature selection approach, that uses a stochastic local search procedure, embedding a naïve Bayes classifier to select an optimal subset of the features constructed in the previous phase. In particular, the optimal subset is searched using a Greedy Randomised Search Procedure (GRASP) [11] and the search is guided by the predictive power of the selected subset computed using a naïve Bayes approach. The focus of this paper is on combining probabilistic feature construction and feature selection for relational sequence learning.

Related works may be divided into two categories. The former includes works belonging to the Inductive Logic Programming (ILP) [12] area, that reformulate the initial relational problem into an attribute-value form, by using frequent patterns as new Boolean features, and then applying propositional learners. The latter category includes all the systems purposely designed to tackle the problem of relational sequence analysis falling into the more specific SRL area where probabilistic models are combined with relational learning.

This work may be related to that in [9], where the authors presented one of the first ILP feature construction methods. They firstly build a set of features adopting a declarative language to constrain the search space and find discriminant features. Then, these features are used to learn a classification model with a propositional learner. In [13] are presented a logic language for mining sequences of logical atoms and an inductive algorithm, that combines principles of the level-wise search algorithm with the version space in order to find all patterns that satisfy a given constraint. These ILP works, however, take into account the feature construction problem only. In this paper, on the other hand, we want to optimise the predictive accuracy of a probabilistic model built on an optimal set of the constructed features.

More similar to our approach are sequential statistical relational techniques that combine a probabilistic model with a relational description belonging to the SRL area, such as Logical Hidden Markov Models (LoHMMs) [3] and relational Conditional Random Fields [4] that are purposely designed for relational sequence learning. In [3] the authors proposed an algorithm for selecting LoHMMs from logical sequences. The proposed logical extension of HMMs overcomes their weakness on flat symbols by handling sequences of structured symbols by means of a probabilistic ILP framework. In [14] the authors presented a method to compute the gradient of the likelihood with respect to the parameters of a LoHMM.

[1] `Lynx` is public available at `http://www.di.uniba.it/~ndm/lynx/`

They overcome the predictive accuracy of the generative model of LoHMMs using a Fisher Kernel. Finally, in [4] an extension of Conditional Random Fields (CRFs) to logical sequences has been proposed. CRFs are undirected graphical models that, instead of learning a generative model as in HMMs, learn a discriminative model designed to handle non-independent input features. In [4], the authors lifted CRFs to the relational case representing the potential functions as a sum of relational regression trees.

2 Lynx: A Relational Pattern-Based Classifier

This section firstly briefly reports the framework for mining relational sequences introduced in [15] and used in Lynx due to its general logic formalism. Over that framework Lynx implements a probabilistic pattern-based classifier. After introducing the representation language, the Lynx system will be presented, along with its feature construction capability, the adopted pattern-based classification model, and the feature selection approach.

As a representation language we used first-order logic.A first-order *alphabet* consists of a set of *constants*, a set of *variables*, a set of *function symbols*, and a non-empty set of *predicate symbols*. Both function symbols and predicate symbols have a natural number (its *arity*) assigned to it. A *term* is a constant symbol, a variable symbol, or an n-ary function symbol f applied to n terms $t_1, t_2, \ldots, t_n$. An atom $p(t_1, \ldots, t_n)$ is a predicate symbol p of arity n applied to n terms t_i. Both l and its negation $\overline{l}$ are said to be (resp., positive and negative) *literals* whenever l is an atom. Literals and terms are said to be *ground* whenever they do not contain variables. A *substitution* θ is defined as a set of bindings $\{X_1 \leftarrow a_1, \ldots, X_n \leftarrow a_n\}$ where $X_i, 1 \leq i \leq n$ are variables and $a_i, 1 \leq i \leq n$ are terms. A substitution θ is applied to an expression e, obtaining the expression $(e\theta)$, by replacing all variables X_i with their corresponding term a_i.

Lynx adopts the relational framework, and the corresponding pattern mining algorithm, reported in [15], that here we briefly recall. Considering a sequence as an ordered succession of events, fluents have been used to indicate that an atom is true for a given event. A *multi-dimensional relational sequence* may be defined as a set of atoms, concerning n dimensions, where each event may be related to another event by means of the $<_i$ operators, $1 \leq i \leq n$. In order to represent multi-dimensional relational patterns, the following dimensional operators have been introduced. Given a set $\mathcal{D}$ of dimensions, $\forall i \in \mathcal{D}$: $<_i$ indicates the direct successor on the dimension i; $\lhd_i$ encodes the transitive closure of $<_i$; and $\bigcirc_i^n$ calculates the n-th direct successor. Hence, a *multi-dimensional relational pattern* may be defined as a set of atoms, regarding n dimensions, in which there are non-dimensional atoms and each event may be related to another event by means of the operators $<_i$, $\lhd_i$ and $\bigcirc_i^n$, $1 \leq i \leq n$. In order to compute the frequency of a pattern over a sequence it is important to define the concept of sequence subsumption. Given $\Sigma = \mathcal{B} \cup U$, where U is the set of atoms in a sequence S, and $\mathcal{B}$ is a background knowledge. A pattern P *subsumes* a sequence S ($P \subseteq S$), iff there exists an SLD_{OI}-deduction of P from Σ. An SLD_{OI}-deduction is an SLD-deduction under Object Identity [16].

2.1 Feature Construction via Pattern Mining

The first step of `Lynx` carries out a feature construction process by mining frequent patterns from sequences with an approach similar to that reported in [9]. The algorithm for frequent multi-dimensional relational pattern mining is based on the same idea as the generic level-wise search method, known in data mining from the Apriori algorithm [17]. The level-wise algorithm performs a breadth-first search in the lattice of patterns ordered by a specialization relation $\preceq$. Generation of the frequent patterns is based on a top-down approach. The algorithm starts with the most general patterns. Then, at each step it tries to specialise all the candidate frequent patterns, discarding the non-frequent patterns and storing those whose length is equal to the user specified input parameter `maxsize`. For each new refined pattern, semantically equivalent patterns are detected, by using the θ_{OI}-subsumption relation [16], and discarded. In the specialization phase the specialization operator, basically, adds atoms to the pattern.

The algorithm uses a background knowledge $\mathcal{B}$ containing the sequences and a set of constraints, similar to that defined in SeqLog [13], that must be satisfied by the generated patterns. In particular, some of the constraints in $\mathcal{B}$ are (see [15] for more details): `maxsize(M)`, maximal pattern length; `minfreq(m)`, the frequency of the patterns must be greater than m; `type(p) and mode(p)`, denote, respectively, the type and the input/output mode of the predicate's arguments `p`, used to specify a language bias; `negconstraint(`$[p_1, p_2, \ldots, p_n]$`)` specifies a constraint that the patterns must not fulfil; `posconstraint(`$[p_1, p_2, \ldots, p_n]$`)` specifies a constraint that the patterns must fulfil; `atmostone(`$[p_1, p_2, \ldots, p_n]$`)` discards all the patterns that make true more than one predicate among p_1, p_2,..., p_n; `key(`$[p_1, p_2, \ldots, p_n]$`)` specifies that each pattern must have one of the predicates $p_1, p_2, \ldots p_n$ as a starting literal.

Given a set of relational sequences D defined over a set of classes C, the *frequency* of a pattern p, $\text{freq}(p, D)$, corresponds to the number of sequences $s \in D$ such that p subsumes s. The *support* of a pattern p with respect to a class $c \in C$, $\text{supp}_c(p, D)$ corresponds to the number of sequences $s \in D$ whose class label is c. Finally, the *confidence* of a pattern p with respect to a class $c \in C$ is defined as $\text{conf}_c(p, D) = \text{supp}_c(p, D)/\text{freq}(p, D)$.

The refinement of patterns is obtained by using a refinement operator ρ that maps each pattern to a set of its specializations, i.e. $\rho(p) \subset \{p' | p \preceq p'\}$ where $p \preceq p'$ means that p is more general than p' or that p subsumes p'. For each specialization level, before starting the next refinement step, `Lynx` records all the obtained patterns. Hence, it might happen that the final set includes a pattern p that subsumes many other patterns in the same set. However, the subsumed patterns may have a different support, contributing in different way to the classification model.

2.2 Pattern-Based Classification

After identifying the set of frequent patterns, the next question is how to use them as features in order to correctly classify unseen sequences. Let $\mathcal{X}$ be the

input space of relational sequences, and $\mathcal{Y} = \{1, 2, \ldots, Q\}$ denote the finite set of possible class labels. Given a training set $D = \{(X_i, Y_i) | 1 \leq i \leq m\}$, where $X_i \in \mathcal{X}$ is a single relational sequence and $Y_i \in \mathcal{Y}$ is the label associated to X_i, the goal is to learn a function $h : \mathcal{X} \rightarrow \mathcal{Y}$ from D that predicts the label for each unseen instance. Let $\mathcal{P}$, with $|\mathcal{P}| = d$, be the set of constructed features obtained in the first step of the `Lynx` system (the patterns mined from D), as reported in Section 2.1. For each sequence $X_k \in \mathcal{X}$ we can build a d-component vector-valued $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ random variable where each $x_i \in \mathbf{x}$ is 1 if the pattern $p_i \in \mathcal{P}$ subsumes sequence X_k, and 0 otherwise, for each $1 \leq i \leq d$.

Using the Bayes' theorem, if $p(Y_j)$ describes the prior probability of class Y_j, then the posterior probability $p(Y_j|\mathbf{x})$ can be computed from $p(\mathbf{x}|Y_j)$ as

$$p(Y_j|\mathbf{x}) = \frac{p(\mathbf{x}|Y_j)p(Y_j)}{\sum_{i=1}^{Q} p(\mathbf{x}|Y_i)p(Y_i)}.$$

Given a set of discriminant functions $g_i(\mathbf{x})$, $i = 1, \ldots, Q$, a classifier is said to assign the vector $\mathbf{x}$ to class Y_j if $g_j(\mathbf{x}) > g_i(\mathbf{x})$ for all $j \neq i$. Taking $g_i(\mathbf{x}) = P(Y_i|\mathbf{x})$, the maximum discriminant function corresponds to the *maximum a posteriori* (MAP) probability. For minimum error rate classification, the following discriminant function will be used

$$g_i(\mathbf{x}) = \ln p(\mathbf{x}|Y_i) + \ln P(Y_i). \tag{1}$$

We are considering a multi-class classification problem involving discrete features. In this problem the components of vector $\mathbf{x}$ are binary-valued and conditionally independent. In particular, let the component of vector $\mathbf{x} = (x_1, \ldots, x_d)$ be binary valued (0 or 1). We define

$$p_{ij} = \text{Prob}(x_i = 1|Y_j)_{\substack{i=1,\ldots,d \\ j=1,\ldots,Q}}$$

with the components of $\mathbf{x}$ being statistically independent for all $x_i \in \mathbf{x}$. In this model each feature x_i gives a yes/no answer about pattern p_i. However, if $p_{ik} > p_{it}$ we expect the i-th pattern to subsume a sequence more frequently when its class is Y_k than when it is Y_t. The factors p_{ij} can be estimated by frequency counts on the training examples, as $p_{ij} = \text{support}_{Y_j}(p_i)$. In this way, the constructed features p_i may be viewed as *probabilistic features* expressing the relevance for pattern p_i in determining classification Y_j.

By assuming conditional independence we can write $P(\mathbf{x}|Y_i)$ as a product of the probabilities of the components of $\mathbf{x}$. Given this assumption, a particularly convenient way of writing the class-conditional probabilities is as follows: $P(\mathbf{x}|Y_j) = \prod_{i=1}^{d} (p_{ij})^{x_i} (1 - p_{ij})^{1-x_i}$. Hence, Eq. 1 yields the discriminant function

$$g_j(\mathbf{x}) = \ln p(\mathbf{x}|Y_j) + \ln p(Y_j) = \sum_{i=1}^{d} x_i \ln \frac{p_{ij}}{1 - p_{ij}} + \sum_{i=1}^{d} \ln(1 - p_{ij}) + \ln p(Y_j). \tag{2}$$

The factor corresponding to the prior probability for class Y_j can be estimated from the training set as $p(Y_i) = \frac{|\{(X,Y) \in D \text{ s.t. } Y = Y_i\}|}{|D|}$, $1 \leq i \leq Q$. The minimum

probability of error is achieved by the following decision rule: decide Y_k, $1 \leq k \leq Q$, if $\forall j, 1 \leq j \leq Q \wedge j \neq k : g_k(\mathbf{x}) \geq g_j(\mathbf{x})$, where $g_i(\cdot)$ is defined as in Eq. 2. Note that this discriminant function is linear in x_i, and thus we can write $g_j(\mathbf{x}) = \sum_{i=1}^{d} \alpha_i x_i + \beta_0$, where $\alpha_i = \ln(p_{ij}/(1-p_{ij}))$, and $\beta_0 = \sum_{i=1}^{d} \ln(1-p_{ij}) + \ln p(Y_j)$. The magnitude of the weight α_i in $g_j(\mathbf{x})$ indicates the relevance of a subsumption for pattern p_i in determining classification Y_j. This is the probabilistic characteristic of the features obtained in the feature construction phase, as opposed to the classical Boolean feature approach.

2.3 Feature Selection with Stochastic Local Search

After having constructed a set of features, and presented a method to use those features to classify unseen sequences, now the problem is how to find a subset of these features that optimises prediction accuracy. The optimization problem of selecting a subset of features (patterns) with a superior classification performance may be formulated as follows. Let $\mathcal{P}$ be the constructed original set of patterns, and let $f : 2^{|\mathcal{P}|} \rightarrow \mathbb{R}$ be a function scoring a selected subset $X \subseteq \mathcal{P}$. The problem of feature selection is to find a subset $\widehat{X} \subseteq \mathcal{P}$ such that $f(\widehat{X}) = \max_{Z \subseteq \mathcal{P}} f(Z)$. An exhaustive approach to this problem would require examining all $2^{|\mathcal{P}|}$ possible subsets of the feature set $\mathcal{P}$, making it impractical for even small values of $|\mathcal{P}|$. The use of a stochastic local search procedure [10] allows to obtain *good* solutions without having to explore the whole solution space.

Given a subset $P \subseteq \mathcal{P}$, for each sequence $X_j \in \mathcal{X}$ we let the classifier find the MAP hypothesis $\widehat{h}_P(X_j) = \arg\max_i g_i(\mathbf{x}_j)$ by adopting the discriminant function reported in Eq. 1, where $\mathbf{x}_j$ is the feature based representation of sequence X_j obtained using patterns in P. Hence the initial optimization problem corresponds to minimise the expectation $\mathrm{E}[\mathbf{1}_{\widehat{h}_P(X_j) \neq Y_j}]$ where $\mathbf{1}_{\widehat{h}_P(X_j) \neq Y_j}$ is the characteristic function of training example X_j returning 1 if $\widehat{h}_P(X_j) \neq Y_j$, and 0 otherwise. Finally, given D the training set with $|D| = m$ and P a set of features (patterns), the number of classification errors made by the Bayesian model is

$$err_D(P) = m\mathrm{E}[\mathbf{1}_{\widehat{h}_P(X_j) \neq Y_j}]. \tag{3}$$

GRASP$^{\mathrm{FS}}$ Consider a *combinatorial optimisation* problem, where one is given a discrete set X of solutions and an objective function $f : X \rightarrow \mathbb{R}$ to be minimised, and seek a solution $x^* \in X$ such that $\forall x \in X : f(x^*) \leq f(x)$. A method to find high-quality solutions for a combinatorial problem consists of a two-step approach made up of a greedy construction phase followed by a perturbative local search [10]. The greedy construction method starts the process from an empty candidate solution and at each construction step adds the best ranked component according to a heuristic selection function. Then, a perturbative local search algorithm, searching a local *neighbourhood*, is used to improve the candidate solution thus obtained. Advantages of this search method are a much better solution quality and fewer perturbative improvement steps needed to reach the local optimum.

GRASP [11] solves the problem of the limited number of different candidate solutions generated by a greedy construction search method by randomising the construction method. GRASP is an iterative process combining at each iteration a construction and a local search phase. In the construction phase a feasible solution is built, and then its neighbourhood is explored by the local search. Algorithm 1 reports the GRASP^{FS} procedure included in the Lynx system to perform the feature selection task. In each iteration, it computes a solution $S \in \mathcal{S}$ by using a randomised constructive search procedure and then applies a local search procedure to S yielding an improved solution. The main procedure is made up of two components: a constructive phase and a local search phase.

Algorithm 1. GRASP^{FS}

Input: D: the training set; $\mathcal{P}$: a set of patterns (features); *maxiter*: maximum number of iterations; $err_D(P)$: the evaluation function (see Eq. 3)
Output: solution $\widehat{S} \subseteq \mathcal{P}$

1: $\widehat{S} = \emptyset$, $err_D(\widehat{S}) = +\infty$
2: iter = 0
3: **while** iter < maxiter **do**
4: α = rand(0,1)
5: $S = \emptyset$; $i = 0$
6: **while** $i < n$ **do**
7: $\mathcal{S} = \{S' | S' = add(S, A)\}$ for each component $A \in \mathcal{P}$ s.t. $A \notin S$
8: $\overline{s} = \max\{err_D(T) | T \in \mathcal{S}\}$
9: $\underline{s} = \min\{err_D(T) | T \in \mathcal{S}\}$
10: RCL $= \{S' \in \mathcal{S} | err_D(S') \leq \underline{s} + \alpha(\overline{s} - \underline{s})\}$
11: select the new S, at random, from RCL
12: $i \leftarrow i + 1$
13: $\mathcal{N} = \{S' \in neigh(S) | err_D(S') < err_D(S)\}$
14: **while** $\mathcal{N} \neq \emptyset$ **do**
15: select $S \in \mathcal{N}$
16: $\mathcal{N} \leftarrow \{S' \in neigh(S) | err_D(S') < err_D(S)\}$
17: **if** $err_D(S) < err_D(\widehat{S})$ **then**
18: $\widehat{S} = S$
19: iter = iter + 1
20: **return** $\widehat{S}$

The constructive search algorithm (lines 4-12) used in GRASP^{FS} iteratively adds a solution component by randomly selecting it, according to a uniform distribution, from a set, named *restricted candidate list* (RCL), of highly ranked solution components with respect to a greedy function $g : \mathcal{S} \rightarrow \mathbb{R}$. The probabilistic component of GRASP^{FS} is characterised by a random choice of one of the best candidates in the RCL. In our case the greedy function g corresponds to the error function $err_D(P)$ previously reported in Eq. 3. In particular, given $err_D(P)$, the heuristic function, and $\mathcal{S}$, the set of feasible solutions, $\underline{s} = \min\{err_D(S) | S \in \mathcal{S}\}$ and $\overline{s} = \max\{err_D(S) | S \in \mathcal{S}\}$ are computed. Then the RCL is defined by including in it all the components S such that $err_D(S) \geq \underline{s} + \alpha(\overline{s} - \underline{s})$.

To improve the solution generated by the construction phase, a local search is used (lines 13-16). It works by iteratively replacing the current solution with a better solution taken from the neighbourhood of the current solution while such a better solution exists. Given $\mathcal{P}$ the set of patterns, in order to build the neighbourhood *neigh(S)* of a solution $S = \{p_1, p_2, \ldots, p_t\} \subseteq \mathcal{P}$, the following operators are exploited:

add: $S \rightarrow S \cup \{p_i\}$ where $p_i \in \mathcal{P} \setminus S$;
replace: $S \rightarrow S \setminus \{p_i\} \cup \{p_k\}$ where $p_i \in S$ and $p_k \in \mathcal{P} \setminus S$.

In particular, given a solution $S \in \mathcal{S}$, the elements of the neighbourhood $neigh(S)$ of S are those solutions that can be obtained by applying an elementary modification (add or replace) to S. Local search starts from an initial solution $S^0 \in \mathcal{S}$ and iteratively generates a series of improving solutions $S^1, S^2, \ldots$. At the k-th iteration, $neigh(S^k)$ is searched for an improved solution S^{k+1} such that $err_D(S^{k+1}) < err_D(S^k)$. If such a solution is found, it becomes the current solution. Otherwise, the search ends with S^k as a local optimum.

3 Experiments

Experiments were conducted on protein fold classification, an important problem in biology. The dataset, already used in [14, 3, 4], is made up of logical sequences of the secondary structure of protein domains. The task is to predict one of the five most populated SCOP folds of alpha and beta proteins (a/b): TIM beta/alpha-barrel (c1), NAD(P)-binding Rossmann-fold domains (c2), Ribosomal protein L4 (c23), Cysteine hydrolase (c37), and Phosphotyrosine protein phosphatases I-like (c55). Overall, the class distribution is 721 sequences for class c1, 360 for c2, 274 for c23, 441 for c37 and 290 for c55. As in [4], we used a round robin approach, treating each pair of classes as a separate classification problem, and the overall classification of an example instance is the majority vote among all pairwise classification problems.

Table 1 reports the experimental results of a 10-fold cross-validated accuracy of `Lynx`. Two experiments have been run choosing confidence levels 0.95 and 1.0. For each experiment, `Lynx` was applied on the same data with and without feature selection. In particular, we run classification on the test instances without applying `GRASP`$^{\texttt{FS}}$ in order to have a baseline accuracy value. Indeed, it turns out

Table 1. Cross-validated accuracy on 10 folds of the data of `Lynx` with and without feature selection

Conf.	`Lynx`	Folds										Mean
		1	2	3	4	5	6	7	8	9	10	
0.95	w/o `GRASP`$^{\texttt{FS}}$	0.84	0.88	0.83	0.83	0.85	0.76	0.85	0.81	0.82	0.80	0.826
	w `GRASP`$^{\texttt{FS}}$	0.88	0.92	0.88	0.88	0.89	0.84	0.93	0,87	0.90	0.93	**0.878**
1.0	w/o `GRASP`$^{\texttt{FS}}$	0.89	0.94	0.84	0.92	0.94	0.88	0.91	0.89	0.88	0.87	0.896
	w `GRASP`$^{\texttt{FS}}$	0.94	0.97	0.93	0.95	0.95	0.93	0.93	0.97	0.90	0.94	**0.942**

Table 2. Cross-validated accuracy of LoHHMs, Fisher kernels, TildeCRF and `Lynx`

System	Accuracy
`LoHMMs` [3]	75%
`Fisher kernels` [14]	84%
`TildeCRF` [4]	92.96%
`Lynx`	**94.15**%

that accuracy grows when $\texttt{GRASP}^{\texttt{FS}}$ optimises the feature set, proving the validity of the method adopted for the feature selection task. Furthermore, the accuracy level grows up when we mine patterns with a confidence level equal to 1.0 which corresponds to saving *jumping emerging patterns*[2] only. This proves that jumping patterns have a discriminative power greater than *emerging patterns* (when the confidence level is equal to 0.95).

As a second experiment we compared `Lynx` on the same data to other SRL systems. Cross-validated accuracy is summarised in Table 2. `LoHHMs` [3] were able to achieve a predictive accuracy of 75%, `Fisher kernels` [14] achieved an accuracy of about 84%, `TildeCRF` [4] reached an accuracy value of 92.96%, while `Lynx` obtained an accuracy of 94.15%. We can conclude that `Lynx` performs better than established methods on this real-world dataset.

4 Conclusions

In this paper we considered the problem of relational sequence learning using relevant patterns discovered from a set of labelled sequences. We firstly apply a feature construction method in order to map each relational sequence into a feature vector. Then, a feature selection algorithm to find an optimal subset of the constructed features leading to high classification accuracy is applied. The performance of the proposed method on a real-world dataset shows an improvement when compared to other sequential statistical relational techniques.

Acknowledgment

This work is partially founded by the MBLab Italian MIUR-FAR project: "The Molecular Biodiversity LABoratory Initiative" (DM19410).

References

1. Kersting, K., De Raedt, L., Gutmann, B., Karwath, A., Landwehr, N.: Relational sequence learning. In: De Raedt, L., Frasconi, P., Kersting, K., Muggleton, S. (eds.) Probabilistic Inductive Logic Programming. LNCS (LNAI), vol. 4911, pp. 28–55. Springer, Heidelberg (2008)

[2] An *emerging pattern* is a pattern whose support in one class differs from its support in others. A *jumping emerging* pattern is a pattern with non-zero support on a class and zero support on all other classes, i.e. with confidence equal to 1.

2. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2007)
3. Kersting, K., De Raedt, L., Raiko, T.: Logical hidden markov models. Journal of Artificial Intelligence Research 25, 425–456 (2006)
4. Gutmann, B., Kersting, K.: TildeCRF: Conditional random fields for logical sequences. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 174–185. Springer, Heidelberg (2006)
5. Antanas, L., Gutmann, B., Thon, I., Kersting, K., De Raedt, L.: Combining video and sequential statistical relational techniques to monitor card games. In: Thurau, C., Driessens, K., Missura, O. (eds.) ICML Workshop on Machine Learning and Games (2010)
6. Kramer, S., Lavrac, N., Flach, P.: Propositionalization approaches to relational data mining. In: Dzeroski, S., Lavrac, N. (eds.) Relational Data Mining, pp. 262–291. Springer, Heidelberg (2001)
7. Dehaspe, L., Toivonen, H., King, R.: Finding frequent substructures in chemical compounds. In: Agrawal, R., Stolorz, P., Piatetsky-Shapiro, G. (eds.) 4th International Conference on Knowledge Discovery and Data Mining, pp. 30–36. AAAI Press, Menlo Park (1998)
8. King, R.D., Srinivasan, A., DeHaspe, L.: Warmr: A data mining tool for chemical data. Journal of Computer-Aided Molecular Design 15(2), 173–181 (2001)
9. Kramer, S., De Raedt, L.: Feature construction with version spaces for biochemical applications. In: Proceedings of the 18th International Conference on Machine Learning, pp. 258–265. Morgan Kaufmann Publisher Inc., San Francisco (2001)
10. Hoos, H., Stützle, T.: Stochastic Local Search: Foundations & Applications. Morgan Kaufmann Publishers Inc., San Francisco (2004)
11. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133 (1995)
12. Muggleton, S., De Raedt, L.: Inductive logic programming: Theory and methods. Journal of Logic Programming 19/20, 629–679 (1994)
13. Lee, S., De Raedt, L.: Constraint based mining of first order sequences in seqLog. In: Meo, R., Lanzi, P., Klemettinen, M. (eds.) Database Support for Data Mining Applications. LNCS (LNAI), vol. 2682, pp. 154–173. Springer, Heidelberg (2004)
14. Kersting, K., Gärtner, T.: Fisher kernels for logical sequences. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 205–216. Springer, Heidelberg (2004)
15. Esposito, F., Di Mauro, N., Basile, T., Ferilli, S.: Multi-dimensional relational sequence mining. Fundamenta Informaticae 89(1), 23–43 (2008)
16. Ferilli, S., Di Mauro, N., Basile, T.M.A., Esposito, F.: θ-subsumption and resolution: A new algorithm. In: Zhong, N., Raś, Z.W., Tsumoto, S., Suzuki, E. (eds.) ISMIS 2003. LNCS (LNAI), vol. 2871, pp. 384–391. Springer, Heidelberg (2003)
17. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the International Conference on Data Engineering, pp. 3–14 (1995)

Learning with Semantic Kernels for Clausal Knowledge Bases

Nicola Fanizzi and Claudia d'Amato

Computer Science Department – University of Bari
{fanizzi|claudia.damato}@di.uniba.it

Abstract. Many applicative domains require complex multi-relational representations. We propose a family of kernels for relational representations to produce statistical classifiers that can be effectively employed in a variety of such tasks. The kernel functions are defined over the set of objects in a knowledge base parameterized on a notion of context, represented by a committee of concepts expressed through logic clauses. A preliminary feature construction phase based on genetic programming allows for the selection of optimized contexts. An experimental session on the task of similarity search proves the practical effectiveness of the method.

1 Statistical Learning for Complex Representations

Many applicative domains, spanning from natural language processing to bio- and chemio-informatics, require complex (multi-)relational representations such as those offered by logic databases (such the *deductive databases*). Standard tasks involving these kinds of knowledge bases require complex forms of inference (e.g. based on a logic calculus) which hardly scale with their dimensions. In such settings, decisions made by exploiting an induced statistical model may represent a viable alternative for supporting related tasks such as (approximate) retrieval, query answering, etc..

Learning inductive classification models for complex knowledge bases can be performed through *Statistical Relational Learning* (SRL) methods. In this work, we intend to adapt efficient non-parametric methods based on kernel functions, originally devised for attribute-value representations, to the multi-relational case required by the mentioned applications. In particular, we will focus on similarity-based methods which are based on density functions which are ultimately grounded on the semantics of the instances of the knowledge bases.

Following the rationale behind the KFOIL system [11], efficient learning methods like the kernel machines [17] may be adapted to work on multi-relational spaces, such as clausal spaces investigated in ILP (and SRL). This required the definition of suitable kernel functions which encode a notion of similarity over such spaces. Even more so, the very kernel function can be the preliminary objective of learning, or measure induction and performance evaluation may be intertwined, as in KFOIL.

Most of the proposed similarity measures for concept descriptions focus on the similarity of atomic concepts within simple concept hierarchies or are strongly based on the structure of the terms for specific FOL fragments. These approaches have been

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 250–259, 2011.

specifically aimed at assessing similarity between concepts [13]. In the perspective of exploiting similarity measures in inductive (instance-based) tasks, the need for a definition of a semantic similarity measure for *instances* arises [15].

Kernel functions may encode a notion of similarity also in the context of structured representations [9]. *Declarative kernels* on mereo-topological instance spaces [7] are supported by a background knowledge made up of a type system and some structural relations (expressing *parthood* and *linkedness*). In some approaches the background knowledge has been partially compiled within kernel machines [8]. Other works have investigated the definition of kernels based on the effect of instance covering [12], or also by considering the similarity of the related proof traces [14].

Kernels for alternative FOL fragments, such as *Description Logics*, have also been proposed. The one defined on the *Feature Description Logic* [2] turned out particularly effective for relational structures elicited from text. More complex description logics have been recently tackled [5]. In this work, a family of declarative kernel functions is proposed that can be applied to knowledge bases expressed in $\mathcal{ALC}$ and $\mathcal{ALCN}$. The kernels encode a notion of similarity of individuals in this representation, based on structural and semantic aspects of the reference representation. Namely a normal form for concept descriptions has been defined which confers the structure of AND-OR trees whose internal nodes contain sub-concepts while the leaves are made up of primitive concepts which can be compared on the grounds of their extension, as elicited from the knowledge base.

The family of functions presented in this work descend more closely from more general kernels which were proposed in [6] which apply to even more complex description logics. They are mainly based on the Minkowski's measures for Euclidean spaces in a way that is similar to the *hypothesis-driven* distances proposed in [16]. Namely, the measures are based on the degree of discernibility of the input objects with respect to a committee of features, which are represented by concept descriptions. As such, these functions depend on both the choice of the feature committee and the knowledge base they are applied to. Differently from the original idea [16], a definition of the notion of projections is given which is based on model-theory for clausal logics.

This leads to investigating on methods for optimizing the committee of features for the measure. To this purpose, the employment of randomized search procedures (and *genetic programming*) may be considered [3]. Experimentally, it may be shown that the measures induced by large committees can be sufficiently accurate when employed for classification tasks even though the employed committee of features were not the optimal ones or if the concepts therein were partially redundant [3]. In the case of kernel machines, this step is performed during the induction of the classifier, thus no *ad hoc* feature construction procedure is needed.

The remainder of the paper is organized as follows. The statistical learning framework is recalled in Sect. 2. The definition of the family of kernels is proposed in Sect. 3, where we prove them to be valid and discuss possible extensions. The effectiveness of the proposed functions is demonstrated with an experimentation, reported in Sect. 4, regarding the task of similarity search. Possible developments are finally examined in Sect. 5.

2 Learning Relational Classifiers through Kernel Methods

Considering the general task of learning classifiers from examples, kernel methods are particularly well suited from an engineering point of view because the learning algorithm (*inductive bias*) and the choice of the kernel function (*language bias*) are almost completely independent [17]. While the former encapsulates the learning task and the way in which a solution is sought, the latter encodes the hypothesis language, i.e. the representation for the target classes. Different kernel functions implement different hypothesis spaces of features. Hence, the same kernel machine can be applied to different representations, provided that suitable kernel functions are available.

Thus, an efficient algorithm may be adopted to work on structured spaces [9] (e.g. trees, graphs) by merely devising a suitable kernel function. Positive and negative examples of the target concept are to be provided to the machine that processes them, through the specific kernel, in order to produce a definition for the target concept in the form of a decision function.

2.1 Clausal Knowledge Bases

In the following, we assume that objects (instances), concepts and relationships among them may be defined in terms of a clausal language such as DATALOG [1], endowed with its standard semantics, which is well suited to support many kind of knowledge bases, such as deductive databases.

A *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$, where $\mathcal{T}$ is a logic *theory* representing the *schema* of the knowledge base, where concepts (entities) and relationships defined through DATALOG clauses, and the *database* $\mathcal{D}$ is a set of ground facts concerning the world state. We will refer to the unary predicates as to concepts. *Primitive* concepts are the atomic concepts defined extensionally by the related facts in $\mathcal{D}$ only, whereas *defined* concepts will be defined by means of clauses contained in $\mathcal{T}$. With $\mathsf{const}(\mathcal{D})$ we denote the set of constants occurring in $\mathcal{K}$ (specifically in $\mathcal{D}$).

As regards inference services, the measures are based on *instance-checking*, which amounts to determining whether an object a belongs to a concept C w.r.t. a given logic interpretation or all the models of the knowledge base: $\mathcal{K} \models C(a)$.

2.2 Learning Linear Classifiers

Working on simple representations, a training example is a vector $\boldsymbol{x}$ of boolean features (propositional variables) extended with an additional one y indicating the membership w.r.t. a target class (i.e. a query concept): $(\boldsymbol{x}, y) \in \{0,1\}^n \times \{-1,+1\}$. Essentially these algorithms aim at finding a vector $\boldsymbol{w} \in \mathbb{R}^n$ which is employed by a linear function to make a decision on the y label for an unclassified instance $(\boldsymbol{x}, \cdot)$: $y(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w} \cdot \boldsymbol{x})$ The resulting rule is: *if* $\boldsymbol{w} \cdot \boldsymbol{x} \geq 0$ *then predict* $\boldsymbol{x}$ *to be positive* $(+1)$ *else it is classified as negative* (-1).

Separating positive from negative instances with a linear boundary may be infeasible as it depends on the complexity of the target concept [17]. The *kernel trick* consists in mapping the examples onto a suitable feature space (likely one with many more dimensions), allowing for the linear separation between positive and negative examples

(*embedding space*. Actually, such a mapping is never explicitly performed; a *valid* (i.e. definite positive) kernel function, corresponding to the inner product of the transformed vectors in the new space, ensures that an embedding exists [17]: $k(x, z) = \phi(x) \cdot \phi(z)$.

Likely, many hyperplanes can separate the examples. Among the other kernel methods, the *support vector machines* (SVMs) aim at finding the hyperplane that maximizes the *margin*, that is the distance from the areas containing positive and negative training examples. The classifier is computed according to the closest instances w.r.t. the boundary (support vectors). These algorithms are very efficient since they solve the problem through quadratic programming techniques once the kernel matrix is produced [17]. The choice of kernel functions is very important as their computation should efficient enough for controlling the complexity of the overall learning process.

2.3 Kernels for Structured Representations

When examples and background knowledge are expressed through structured (logical) representations a further level of complexity is added. One way to solve the problem may involve the transformation of statistical classifiers into logical ones. However while the opposite mapping has been shown as possible, direct solutions to the learning problem are still to be investigated. An appealing quality of the class of valid kernel functions is its closure w.r.t. many operations. In particular this class is closed w.r.t. the *convolution* [17]:

$$\kappa_{\text{conv}}(x, z) = \sum_{\overline{x} \in R^{-1}(x)} \sum_{\overline{z} \in R^{-1}(z)} \prod_{i=1}^{D} \kappa_i(\overline{x}_i, \overline{z}_i)$$

where relationship R builds a single compound out of D simpler objects, each from a space that is already endowed with a valid kernel (κ_i). The choice of R is a non-trivial task as it may depend on the particular application.

Then new kernels can be defined for complex structures based on simpler kernels defined for their parts using the closure property w.r.t. this operation and many others [17]. Many definitions have exploited this property, introducing kernels for strings, trees, graphs and other discrete structures. In particular, the framework in [9] shows a principled way for defining new kernels based on type construction, where types are specified in a declarative way.

3 Semantic Kernels for Clausal Spaces

3.1 Kernel Definition

It can be observed that, although instances seem to lack of a syntactic structure that may be exploited for a comparison, moving to a semantic level, similar objects should *behave* similarly with respect to the same concepts, i.e. similar instantiations should be shared and dissimilar objects should likely instantiate disjoint concepts [3].

Following this rationale, we introduce novel kernel functions for the target representation, that simply compare the semantics of the instances w.r.t. a fixed number

of dimensions represented by concept definitions. Namely, they are compared on the grounds of their behavior w.r.t. a reduced (yet not necessarily disjoint) committee of features, represented by a collection of concepts, say $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, which stands as a group of discriminating *features* expressed in the language taken into account. In this case, we will consider unary predicates which have a definition in the knowledge base i.e. the related set of clauses in terms of the predicates in $\mathcal{K}$. Then, a family of semantic similarity measures for objects can be defined for clausal representations, with a simple formulation, inspired by Minkowski's metrics:

Definition 3.1 (family of kernel functions). *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ be a knowledge base. Given a set of concepts $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$ defined in terms of $\mathcal{K}$, a family $\{\kappa_p^{\mathsf{F}}\}_{p \in \mathbb{N}}$ of functions $\kappa_p^{\mathsf{F}} : \mathsf{const}(\mathcal{D}) \times \mathsf{const}(\mathcal{D}) \mapsto \mathbb{R}$ is defined as follows:*
$\forall a, b \in \mathsf{const}(\mathcal{D})$

$$\kappa_p^{\mathsf{F}}(a, b) := \left[\sum_{i=1}^{|\mathsf{F}|} (\delta(\pi_i(a), \pi_i(b)))^p \right]^{1/p}$$

where δ is the Kronecker symbol *acting as an indicator function and the i-th* projection function *π_i, with $i = 1, \ldots, m$, is defined by:*
$\forall a \in \mathsf{const}(\mathcal{D})$

$$\pi_i(a) = \begin{cases} 1 & \mathcal{K} \vdash F_i(a) \\ 0 & \textit{otherwise} \end{cases}$$

The superscript F or the subscript p will be omitted when fixed.

3.2 Discussion

Primarily, it must be shown that these functions are valid kernels [17].

Proposition 3.1 (validity). *For a fixed feature set F and $p \in \mathbb{N}$, the function κ_p^{F} is a valid kernel.*

Proof. The property can be easily proved considering that the function is defined as a combination of a number of operators to δ, which can be regarded as a simple matching kernel function. The validity descends from the properties of closure of the class of kernel functions w.r.t. those operations. □

We make the assumption that the feature-set F may represent a sufficient number of (possibly redundant) features that are able to discriminate really different objects. As hinted in [16], redundancy may help appreciate the relative differences in similarity.

Compared to other proposed similarity (or dissimilarity) measures, the presented functions are not based on structural (syntactical) criteria; they require only deciding (proof-theoretically) whether an object can be an instance of the concepts in the committee.

Note that the computation of projection functions can be performed in advance (with the support of a suitable DBMS) thus determining a speed-up in the actual computation of the kernel.

3.3 Extensions

In some cases it may be convenient to work with a normalized version of the functions that can be defined as $\bar{\kappa}_p^{\mathsf{F}} : \mathsf{const}(\mathcal{D}) \times \mathsf{const}(\mathcal{D}) \mapsto [0,1]$ such that: $\forall a, b \in \mathsf{const}(\mathcal{D})$

$$\bar{\kappa}_p^{\mathsf{F}}(a,b) := \left[\sum_{i=1}^{|\mathsf{F}|} \left(\frac{\delta(\pi_i(a), \pi_i(b))}{|\mathsf{F}|} \right)^p \right]^{1/p}$$

Alternatively, the standard normalized version of the kernel function κ_p^{F} can be obtained as follows [17]:
$\forall a, b \in \mathsf{const}(\mathcal{D})$

$$\bar{\kappa}_p^{\mathsf{F}}(a,b) := \kappa_p^{\mathsf{F}}(a,b) / \sqrt{\kappa_p^{\mathsf{F}}(a,a) \cdot \kappa_p^{\mathsf{F}}(b,b)}$$

The definition above might be further extended by recurring to model theory. Namely, the set $\mathcal{M}_{\mathcal{K}} \subseteq 2^{|\mathcal{B}_{\mathcal{K}}|}$ of the Herbrand models of the knowledge base can be considered, where $\mathcal{B}_{\mathcal{K}}$ stands for its Herbrand base. Now, given two instances a and b to be compared w.r.t. a certain feature F_i, $i = 1, \ldots, m$, one might check whether they are similar in the world represented by a Herbrand interpretation $\mathcal{I} \in \mathcal{M}_{\mathcal{K}}$: $\mathcal{I} \models F_i(a)$ and $\mathcal{I} \models F_i(b)$.

Hence, a similarity measure should count the cases of agreement, varying the Herbrand models of the knowledge base. The resulting definition for a new kernel function is the following:
$\forall a, b \in \mathsf{const}(\mathcal{D})$

$$\kappa_p^{\mathsf{F}}(a,b) := \frac{1}{|\mathcal{M}_{\mathcal{K}}|} \left[\sum_{\mathcal{I} \in \mathcal{M}_{\mathcal{K}}} \sum_{i=1}^{m} \left(\delta(\pi_i^{\mathcal{I}}(a), \pi_i^{\mathcal{I}}(b)) \right)^p \right]^{1/p}$$

where the projections are computed for a specific world state as encoded by a Herbrand interpretation $\mathcal{I}$:
$\forall a \in \mathsf{const}(\mathcal{D})$

$$\pi_i^{\mathcal{I}}(a) = \begin{cases} 1 & F_i(a) \in \mathcal{I} \\ 0 & \textit{otherwise} \end{cases}$$

Although the measures could be implemented according to the definitions, their effectiveness and also the efficiency of their computation strongly depends on the choice of the feature committee (*feature selection*). Indeed, various optimizations of the measures can be foreseen as concerns their parametric definition. Among the possible committees, those that are able to better discriminate the objects in the dataset ought to be preferred. Finding good committees can be accomplished by means of randomized optimization techniques especially when many examples are available [3]. Namely, part of the entire data can be drawn in order to learn optimal F sets, in advance with respect to the successive usage for all other purposes.

3.4 Classification and Query Answering

The ultimate aim is to apply the classifiers produced by means of kernel methods to clausal knowledge bases to solve practical problems.

As concerns the mere classification and also query answering tasks, the adoption of an inductive rather than deductive method brings various advantages. The learning algorithms for inducing the classifier are polynomial on the size of training samples provided; linear classification is even more straightforward. The induced classification models can be considered as a sort of view on the knowledge base, that can be stored and re-used depending on the specific task or problem to be solved. Most importantly, the kernel-machine (a SVM in our case) is able to learn a *soft-margin* classifier which can perform quite well even with *noisy* examples [17].

The adopted kernel method will not require a particular representation of the training instances: It will be sufficient to provide the related Gram matrix. In our case, given a query concept Q for which an inductive hypothesis function h_Q (a classification model) is to be constructed, each training instance x_i must have a known class-membership value w.r.t. Q, $y_i = t_Q(x_i) \in \{-1, +1\}$, i.e. the true value of the function t_Q that the inductive method aims at approximating with h_Q. Once the kernel machine has learned the hypothesis h_Q (based on a vector of coefficients for the primal or dual form), this will be used for predict classification (and then also for query answering) for the whole population of objects in the knowledge base.

4 Experiments

In order to prove the effectiveness of the kernel functions (coupled with the committee-optimization procedure), an experimentation was performed on the task of query answering based on inductive classification, i.e. finding instances that can be answers to a query by means of hypotheses learned using kernel functions. A comparison can be made to an analogous experiment with the related similarity-based learning method (*nearest-neighbors*) presented in [3].

4.1 Setup

The family relational kernels defined in the previous section has been implemented and integrated with the Java version of the library of SVM algorithms[1] LIBSVM 2.89. The kernel matrix is provided by a suitable Java interface to Prolog reasoning based on external libraries[2]. In the experiments the default values for the parameters of the library were used.

Relational knowledge bases from different domains have been selected analogously to the previous experiment [3]: a small one that was artificially generated for the studying the PHASE TRANSITION, (problem `pt4444`), the University of Washington CSE dataset (UW-CSE), one from the MUTAGENESIS datasets, and one concerning the layout structure of scientific papers (SCI-DOCS). Moreover, in this new experiment also the E_COLI dataset was employed. Further details on these datasets are reported in Tab. 1.

In the experiment, given a number of random query concepts, we intended to assess the accuracy of the answers obtained from the model induced by the kernel method

[1] Publicly available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
[2] JPL 3. See `http://www.swi-prolog.org`

Table 1. Details about the datasets employed

dataset	*#concepts*	*#relations*	*#constants*
PHASE TRANS.	2	4	400
UW-CSE	9	20	2208
SCI-DOCS	30	9	4585
MUTAGENESIS	68	2	9292
E_COLI	224	245	396

Table 2. Experimental results using the models induced by the kernel methods: cardinality of the induced feature committee and average outcomes ($\pm$ standard deviation) over the 10 folds

dataset	F	*%correct*	*%false pos.*	*%false neg.*
PHASE TRANS.	6	98.95 ± 0.13	01.02 ± 0.04	00.03 ± 0.11
UW-CSE	9	98.42 ± 2.65	01.58 ± 2.65	00.00 ± 0.00
SCI-DOCS	7	98.73 ± 5.32	01.27 ± 5.32	00.00 ± 0.00
MUTAGENESIS	11	97.51 ± 2.34	02.49 ± 2.34	00.00 ± 0.00
E_COLI	10	99.22 ± 0.85	00.78 ± 0.85	00.00 ± 0.00

compared to the correct ones, which can be derived by (deductive) reasoning with the knowledge base. For each dataset, a preliminary phase was devoted to the construction of an optimal feature set with the stochastic search method presented in [3]. To this purpose, a limited set of instances (100) was randomly selected in order to perform the task. A simple discretization algorithm had to be preliminarily applied to the numerical attributes, if any, since currently the kernel functions do not handle these cases. Hence, the number of predicates increased w.r.t. the original dataset.

A number of 20 query concepts were random generated in terms of the predicates of the knowledge base. The experiment was repeated applying a ten-fold cross-validation design to each dataset. In the *training phase* classification models were generated using a SVM on the kernel matrices obtained for each dataset. Then, in the *test phase*, the class-membership of all the other instances w.r.t. the query concepts was tested, comparing the inductive prediction to the true value deduced from the knowledge base.

4.2 Outcomes

Considering the outcomes of the experiments shown in Tab. 2 (to be compared to those of the previous experiment in Tab. 3), we note that the performance is generally quite good and we do not observe the decay of the previous experiment for the case of the SCI-DOCS dataset, which also determined the largest variance. Generally, the new method performs comparably well w.r.t. the previous one with some slight decay, however it seems to be much more stable.

As regards the errors made, seldom false negatives were observed with the new method. Thus, we may conclude that the inductive classification appears weaker in terms of recall rather than precision. The good results were probably due to the regularity of the information in the various datasets: for each individual the same amount of

Table 3. Experimental results obtained with a nearest-neighbor procedure: cardinality of the induced feature committee and average outcomes ($\pm$ standard deviation) over the 10 folds

dataset	F	*%correct*	*%false pos.*	*%false neg.*
PHASE TRANS.	6	99.97 ± 0.13	00.00 ± 0.00	00.03 ± 0.13
UW-CSE	9	99.01 ± 1.92	00.05 ± 0.08	00.94 ± 1.94
SCI-DOCS	5	85.49 ± 9.06	01.66 ± 1.87	12.85 ± 8.96
MUTAGENESIS	11	98.68 ± 1.92	00.08 ± 0.12	01.24 ± 1.94

information is known, which helps to discern among them. More sparsity (incomplete information) in the datasets would certainly decrease the discernibility of the objects and, hence, the overall performance.

The good performance on such datasets, despite some of them are known to be particularly hard to learn, is due to the fact that the system actually does not have to learn a definition for an unknown concept (nor a generative model), but, rather, those features that can help to discern between positive and negative instances.

It is also possible to compare the number of new features constructed for the kernel function and the distance measure in the two experiments and the overall number of (primitive or defined) concepts in the knowledge base. As expected, the number of concepts in the committees is also similar although the stochastic search started from random concepts. Employing smaller committees (with comparable performance results) is certainly desirable for the sake of an efficient computation of the measure. However, some of them were generated during the discretization process.

We can conclude that we have the new method is more stable and less error prone w.r.t. the false negatives. Although model construction is not performed in nearest neighbor learning, the kernel-based method is also generally more efficient in the classification phase because the calculation of a distance requires much more reasoning (also depending on the number of neighbors) than the single evaluation of the kernel function.

5 Concluding Remarks and Outlook

In the line of past works on distance-induction, we have proposed the definition of a family of kernel functions over the instances in a clausal knowledge base. The kernels are parameterized on a committee of concepts that can be selected by the proposed randomized search method. An experimentation on performing semantic-based retrieval proved the effectiveness of the new measures.

Possible subsumption relationships between predicates in the committee may be explicitly exploited in the measure for making the relative distances more accurate. The extension to the case of concept distance may also be improved. Furthermore, the measure should be extended to cope with numeric information which abound in biological or chemical datasets.

This kind of measures may have a wide range of applications on clausal knowledge bases. They have been also integrated in an instance-based learning system implementing a nearest-neighbor learning algorithm similar to RIBL [4].

Currently we are exploiting the measures in conceptual clustering algorithms where clusters will be formed by grouping instances on the grounds of their similarity assessed through a distance measure for multi-relational representations [10] which is based on the same rationale of the presented kernel functions, triggering the induction of new emerging concepts. Moreover, we are investigating the application of the same ideas to ontology mining tasks which require specific solution for the peculiar knowledge representations adopted in that context.

References

1. Ceri, S., Gottlob, G., Tanca, L.: Logic Programming and Databases. Springer, Heidelberg (1990)
2. Cumby, C., Roth, D.: On kernel methods for relational learning. In: Fawcett, T., Mishra, N. (eds.) Proc. of ICML 2003, pp. 107–114. AAAI Press, Menlo Park (2003)
3. d'Amato, C., Fanizzi, N., Esposito, F.: Induction of optimal semantic semi-distances for clausal knowledge bases. In: Blockeel, H., Ramon, J., Shavlik, J., Tadepalli, P. (eds.) ILP 2007. LNCS (LNAI), vol. 4894, pp. 29–38. Springer, Heidelberg (2008)
4. Emde, W., Wettschereck, D.: Relational instance-based learning. In: Saitta, L. (ed.) Proc. of ICML 1996, pp. 122–130. Morgan Kaufmann, San Francisco (1996)
5. Fanizzi, N., d'Amato, C., Esposito, F.: Learning with kernels in description logics. In: Železný, F., Lavrač, N. (eds.) ILP 2008. LNCS (LNAI), vol. 5194, pp. 210–225. Springer, Heidelberg (2008)
6. Fanizzi, N., d'Amato, C., Esposito, F.: Statistical learning for inductive query answering on OWL ontologies. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 195–212. Springer, Heidelberg (2008)
7. Frasconi, P., Passerini, A., Muggleton, S., Lodhi, H.: Declarative kernels. Technical Report 2/2004, Dipartimento di Sistemi e Informatica, Università di Firenze (2004)
8. Fung, G., Mangasarian, O., Shavlik, J.: Knowledge-based nonlinear kernel classifiers. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 102–113. Springer, Heidelberg (2003)
9. Gärtner, T., Lloyd, J., Flach, P.: Kernels and distances for structured data. Machine Learning 57(3), 205–232 (2004)
10. Kirsten, M., Wrobel, S.: Relational distance-based clustering. In: Page, D. (ed.) ILP 1998. LNCS (LNAI), vol. 1446, pp. 261–270. Springer, Heidelberg (1998)
11. Landwehr, N., Passerini, A., De Raedt, L., Frasconi, P.: kFOIL: Learning simple relational kernels. In: Proc. of AAAI 2006. AAAI Press, Menlo Park (2006)
12. Muggleton, S., Lodhi, H., Amini, A., Sternberg, M.: Support vector inductive logic programming. In: Hoffmann, A., Motoda, H., Scheffer, T. (eds.) DS 2005. LNCS (LNAI), vol. 3735, pp. 163–175. Springer, Heidelberg (2005)
13. Nienhuys-Cheng, S.-H.: Distances and limits on herbrand interpretations. In: Page, D. (ed.) ILP 1998. LNCS, vol. 1446, pp. 250–260. Springer, Heidelberg (1998)
14. Passerini, A., Frasconi, P., De Raedt, L.: Kernels on Prolog proof trees: Statistical learning in the ILP setting. Journal of Machine Learning Research 7, 307–342 (2006)
15. Ramon, I., Bruynooghe, M.: A framework for defining distances between first-order logic objects. TR CW 263. Dept. of Computer Science, Katholieke Universiteit Leuven (1998)
16. Sebag, M.: Distance induction in first order logic. In: Džeroski, S., Lavrač, N. (eds.) ILP 1997. LNCS(LNAI), vol. 1297, pp. 264–272. Springer, Heidelberg (1997)
17. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)

Topic Graph Based Non-negative Matrix Factorization for Transfer Learning

Hiroki Ogino and Tetsuya Yoshida

Graduate School of Information Science and Technology,
Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
{hiroki,yoshida}@meme.hokudai.ac.jp

Abstract. We propose a method called Topic Graph based NMF for Transfer Learning (TNT) based on Non-negative Matrix Factorization (NMF). Since NMF learns feature vectors to approximate the given data, the proposed approach tries to preserve the feature space which is spanned by the feature vectors to realize transfer learning. Based on the learned feature vectors in the source domain, a graph structure called topic graph is constructed, and the graph is utilized as a regularization term in the framework of NMF. We show that the proposed regularization term corresponds to maximizing the similarity between topic graphs in both domains, and that the term corresponds to the graph Laplacian of the topic graph. Furthermore, we propose a learning algorithm with multiplicative update rules and prove its convergence. The proposed method is evaluated over document clustering problem, and the results indicate that the proposed method improves performance via transfer learning.

1 Introduction

As the amount of available data increases, various research efforts have been conducted to learn knowledge from data. It would be nice if the obtained knowledge learned in one domain can also be utilized in another domain to improve performance in the latter domain. Transfer Learning is a machine learning framework to realize this goal [11]. The domain in which the knowledge is learned is called source domain, and the other domain is called target domain in this paper. Various methods have been proposed to realize transfer learning [3,10,15].

We propose a transfer learning method based on Non-negative Matrix Factorization (NMF). Since NMF learns feature vectors to approximate the given data, the proposed method tries to preserve the feature space which is spanned by the feature vectors in transfer learning. Based on the learned feature vectors in the source domain, a graph structure called topic graph is constructed, and the graph is utilized as a regularization term in the framework of NMF. We show that the proposed regularization term corresponds to maximizing the similarity between topic graphs in both domains, and that the term corresponds to the graph Laplacian [13] of the topic graph. Furthermore, we propose a learning algorithm with multiplicative update rules and prove its convergence.

Once the appropriate representation in the target domain is learned, various algorithms can be applied to the learned representation as in [8]. Especially,

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 260–269, 2011.

contrary to other method [10,15], no "label" information is required to conduct transfer learning in our approach. The proposed method is evaluated over document clustering problem, and the results indicate that the proposed method improves performance via transfer learning. Especially, the proposed method showed large performance improvement even when many features were utilized.

Section 2 explains the details of our method. Section 3 reports the evaluation of the proposed approach over various datasets and discusses the obtained results. Section 4 summarizes our contributions.

2 Topic Graph based NMF for Transfer Learning

We use a bold capital letter for a matrix, and a lower italic letter for a vector. $\mathbf{X}_{ij}$ stands for the element in a matrix $\mathbf{X}$. tr stands for the trace of a matrix, and $\mathbf{X}^T$ stands for the transposition of $\mathbf{X}$.

2.1 Non-negative Matrix Factorization

Under the specified number of features q, Non-negative Matrix Factorization (NMF) [9] factorizes a non-negative matrix $\mathbf{X}=[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \mathbb{R}_+^{p \times n}$ into two non-negative matrices $\mathbf{U}=[\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q] \in \mathbb{R}_+^{p \times q}$, $\mathbf{V}=[\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n] \in \mathbb{R}_+^{q \times n}$ as

$$\mathbf{X} \simeq \mathbf{UV} \tag{1}$$

Each data $\boldsymbol{x}$ is approximated as a linear combination of $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q$. Thus, NMF conducts dimensionality reduction of a data matrix $\mathbf{X}$ and learns the representation $\mathbf{V}$ in the feature space which is spanned by the column vectors of $\mathbf{U}$. Minimization of the following objective function is conducted to obtain the matrices $\mathbf{U}$ and $\mathbf{V}$:

$$J_1 = ||\mathbf{X} - \mathbf{UV}||^2 \tag{2}$$

where $||\cdot||$ stands for the norm of a matrix. In this paper we focus on Frobenius norm $||\cdot||_F$ [9].

In most approaches which utilize NMF for document clustering, the number of features are set to the number of clusters [14,5], and each instance is assigned to the cluster c with the maximal value in the constructed representation $\boldsymbol{v}$.

$$c = \underset{c}{\operatorname{argmax}}\, v_c \tag{3}$$

where v_c stands for the value of c-th element in $\boldsymbol{v}$.

2.2 Topic Graph

We utilize NMF for realizing transfer learning. With NMF, each data $\boldsymbol{x}$ is approximated as a linear combination of $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_q$ in eq.(1). Thus, we regard each column vector of $\mathbf{U}$ as a topic.

Furthermore, the proposed method constructs a graph structure called topic graph from the learned $\mathbf{U}$ and utilizes the graph to realize transfer learning. The topic graph is defined by mapping each topic (column vector of $\mathbf{U}$) to a vertex and connecting each pair of vertices with their similarities. Currently cosine similarity is utilized to define the edge weights in the topic graph as:

$$\mathbf{W} = \mathbf{U}^T\mathbf{U}, \qquad where \;\; \boldsymbol{u}_l^T\boldsymbol{u}_l = 1, \quad \forall l = 1, \ldots, q \tag{4}$$

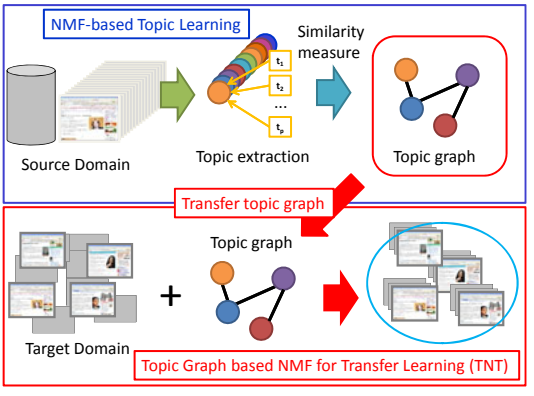


Fig. 1. Overview of Topic based NMF for Transfer Learning

Step1 apply NMF on $\mathbf{X}_s$ in the source domain and obtain $\mathbf{U}_s$
Step2 transfer the learned $\mathbf{U}_s$ into the target domain
  Step2.1 construct the topic graph $\mathbf{W}_s$ from $\mathbf{U}_s$
  Step2.2 learn $\mathbf{V}_t$ in the target domain by topic-graph based NMF (Algorithm 1)
Step3 apply some learning algorithm to $\mathbf{V}_t$

Fig. 2. The Framework of Topic based NMF for Transfer Learning

2.3 Overview

The overview of the proposed approach is illustrated in Fig. 1. Since NMF learns $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_q]$ which constitutes the feature space $Span(\mathbf{U})$ [6], based on our transfer hypothesis that feature spaces in both domains are similar, the proposed method conducts transfer learning by preserving the feature space when the data matrix $\mathbf{X}_t$ in the target domain is factorized. The topic graph based on $\mathbf{U}_s$ in eq.(4) is utilized as a regularization term to conduct transfer learning in the proposed algorithm.

The framework of the proposed transfer learning is summarized in Fig. 2. By applying NMF to the data matrix $\mathbf{X}_s$ in the source domain, $\mathbf{U}_s$ is learned and transfered to the target domain in Step 1. The topic graph $\mathbf{W}_s$ in eq.(4) is constructed from $\mathbf{U}_s$ in Step 2.1, and $\mathbf{V}_t$ in the target domain is learned by the proposed algorithm TNT (in Algorithm 1) in Step2.2. Finally, some learning algorithm is applied to the learned representation $\mathbf{V}_t$ as in [8].

2.4 Topic Graph based NMF for Transfer Learning

As explained above, our transfer hypothesis is the similarity (preservation) of feature spaces in both domains, i.e., $Span(\mathbf{U}_s) \simeq Span(\mathbf{U}_t)$ based on $\mathbf{U}_s$ and $\mathbf{U}_t$

in NMF. This constraint can be formalized as the minimization of $||\mathbf{U}_s - \mathbf{U}_t||^2$ with respect to the norm of matrices as in eq.(2). Thus, if each column vector in both $\mathbf{U}_s$ and $\mathbf{U}_t$ is normalized, this constraint can be formalized based on the non-negativity in NMF as follows:

$$\begin{aligned}
\underset{\mathbf{U}_t}{\operatorname{argmin}} \ ||\mathbf{U}_s - \mathbf{U}_t||^2 &\Leftrightarrow \underset{\mathbf{U}_t}{\operatorname{argmax}} \ \operatorname{tr}(\mathbf{U}_s\mathbf{U}_t^T) && (5)\\
&\Leftrightarrow \underset{\mathbf{U}_t}{\operatorname{argmax}} \ \operatorname{tr}((\mathbf{U}_s\mathbf{U}_t^T)^T(\mathbf{U}_s\mathbf{U}_t^T)) && (6)\\
&\Leftrightarrow \underset{\mathbf{U}_t}{\operatorname{argmin}} \operatorname{tr}(\mathbf{U}_t(\mathbf{D}_s - \mathbf{W}_s)\mathbf{U}_t^T) && (7)\\
&\Leftrightarrow \underset{\mathbf{U}_t}{\operatorname{argmin}} \operatorname{tr}(\mathbf{U}_t\mathbf{L}_s\mathbf{U}_t^T) && (8)
\end{aligned}$$

where $\mathbf{D}_s$ in eq.(7) is the degree matrix of the topic graph $\mathbf{W}_s$, $\mathbf{L}_s = \mathbf{D}_s$ - $\mathbf{W}_s$ is the graph Laplacian for $\mathbf{W}_s$. Eq.(5) and eq. (6) are equivalent due to the non-negativity of $\mathbf{U}_s$ and $\mathbf{U}_t$. Since $\operatorname{tr}(\mathbf{U}_t\mathbf{D}_s\mathbf{U}_t^T)$ is some constant, eq.(6) and eq. (7) are equivalent.

Thus, the proposed method formalizes transfer learning as the minimization of the following objective function based on $\mathbf{U}_s$.

$$J_2 = ||\mathbf{X}_t - \mathbf{U}_t\mathbf{V}_t||^2 + \nu \operatorname{tr}(\mathbf{U}_t\mathbf{L}_s\mathbf{U}_t^T) \tag{9}$$

Furthermore, minimization of the second term in eq.(9) can be shown as equivalent to the following:

$$\begin{aligned}
\underset{\mathbf{U}_t}{\operatorname{argmax}} \ \operatorname{tr}(\mathbf{U}_t(\mathbf{U}_s^T\mathbf{U}_s)\mathbf{U}_t^T) &\Leftrightarrow \underset{\mathbf{U}_t}{\operatorname{argmax}} \ \operatorname{tr}(\mathbf{W}_s\mathbf{W}_t) && (10)\\
&\Leftrightarrow \underset{\mathbf{U}_t}{\operatorname{argmax}} \ \mathbf{W}_s \bullet \mathbf{W}_t && (11)
\end{aligned}$$

where $\bullet$ represents the inner product of matrices [6]D Thus, the proposed second term in eq.(9) corresponds to maximizing the similarity of topic graphs $\mathbf{W}_s$ and $\mathbf{W}_t$ in both domains. Furthermore, from eq. (8), it can also considered as the regularization term based on the graph Laplacian of the proposed topic graph.

2.5 The Algorithm

We propose an algorithm to find out $\mathbf{U}_t$ and $\mathbf{V}_t$ by minimizing eq.(9). By introducing Lagrangian multipliers $\boldsymbol{\Psi}, \boldsymbol{\Phi}$ for the non-negativity of each element in $\mathbf{U}_t$ and $\mathbf{V}_t$, we define the following Lagrangian eq.(12):

$$\mathcal{L} = \underset{\mathbf{U}_t,\mathbf{V}_t}{\operatorname{argmin}} ||\mathbf{X}_t - \mathbf{U}_t\mathbf{V}_t||^2 + \nu \operatorname{tr}(\mathbf{U}_t\mathbf{L}_s\mathbf{U}_t^T) + \operatorname{tr}(\boldsymbol{\Psi}\mathbf{U}_t^T) + \operatorname{tr}(\boldsymbol{\Phi}\mathbf{V}_t^T) \tag{12}$$

By the partial derivative of eq.(12) w.r.t. $\mathbf{U}_t$ and $\mathbf{V}_t$, and using the KKT (Karush-Kuhn-Tucker) condition, we can derive the following multiplicative update rules

Algorithm 1. Topic based NMF for Transfer Learning

Algorithm TNT($\mathbf{X}_t$, $\mathbf{U}_s$, ν)

Require: $\mathbf{X}_t$
Require: $\mathbf{U}_s \in \mathbb{R}_+^{p \times q}$ s.t. $\boldsymbol{u}_l^T \boldsymbol{u}_l = 1, \quad \forall l = 1, \ldots, q$
Require: $\nu \in \mathbb{R}_+$ *// regularization parameter*
1: $\mathbf{U}_t := \mathbf{U}_s$ *//utilize* $\mathbf{U}_s$ *for initialization*
2: initialize $\mathbf{V}_t$
3: **while** termination condition is not satisfied **do**
4: $(\mathbf{U}_t)_{ij} := (\mathbf{U}_t)_{ij} \frac{(\mathbf{X}_t \mathbf{V}_t^T + \nu \mathbf{U}_t \mathbf{W}_s)_{ij}}{(\mathbf{U}_t \mathbf{V}_t \mathbf{V}_t^T + \nu \mathbf{U}_t \mathbf{D}_s)_{ij}}$
5: normalize $\mathbf{U}_t$ s.t. $\boldsymbol{u}_l^T \boldsymbol{u}_l = 1, \quad \forall l = 1, \ldots, q$
6: $(\mathbf{V}_t)_{ij} := (\mathbf{V}_t)_{ij} \frac{(\mathbf{U}_t^T \mathbf{X}_t)_{ij}}{(\mathbf{U}_t^T \mathbf{U}_t \mathbf{V}_t)_{ij}}$
7: **end while**
8: **return** $\mathbf{V}_t$

$$(\mathbf{U}_t)_{ij} \leftarrow (\mathbf{U}_t)_{ij} \frac{(\mathbf{X}_t \mathbf{V}_t^T + \nu \mathbf{U}_t \mathbf{W}_s)_{ij}}{(\mathbf{U}_t \mathbf{V}_t \mathbf{V}_t^T + \nu \mathbf{U}_t \mathbf{D}_s)_{ij}} \tag{13}$$

$$(\mathbf{V}_t)_{ij} \leftarrow (\mathbf{V}_t)_{ij} \frac{(\mathbf{U}_t^T \mathbf{X}_t)_{ij}}{(\mathbf{U}_t^T \mathbf{U}_t \mathbf{V}_t)_{ij}} \tag{14}$$

TNT (Topic based NMF for Transfer Learning) is shown in Algorithm 1.

The following theorem holds for algorithm TNT:

Theorem 1. *The objective function in eq.(9) is non-increasing under the update rules in Algorithm 1, and since it is non-negative, the algorithm converges.*

The proof based on the auxiliary function [1] is omitted due to space limitation.

2.6 Regularization via Pairwise Relation in the Target Domain

In order to improve the performance in the target domain, we propose the following objective function by adding another regularization term in eq.(9) based on pairwise data relation [1]:

$$J_3 = ||\mathbf{X}_t - \mathbf{U}_t \mathbf{V}_t||^2 + \nu \operatorname{tr}(\mathbf{U}_t \mathbf{L}_s \mathbf{U}_t^T) + \lambda \operatorname{tr}(\mathbf{V}_t^T \mathbf{L}_t \mathbf{V}_t) \tag{15}$$

The third term corresponds to the regularization term in [1], and λ is the regularization parameter. In [1], the m nearest neighbor (m-NN) graph in data space is first constructed, and the graph Laplacian $\mathbf{L} = \mathbf{D}$ - $\mathbf{A}$ for the adjacency matrix $\mathbf{A}$ of the graph is utilized. In our method, the graph Laplacian $\mathbf{L}_t$ for the weight matrix of the m-NN graph is utilized instead.

Even when eq.(15) is utilized, we can also derive the corresponding multiplicative update rules as in eq.(13) and eq.(14). Since the partial derivative of $\mathbf{U}_t$ and that of $\mathbf{V}_t$ are independent, the same update rule in eq.(13) can be used for $\mathbf{U}_t$; the update rule in [1] can be utilized for $\mathbf{V}_t$ to minimize (15). Furthermore we can also show the same convergence theorem as in Theorem 1.

Table 1. 4 clusters dataset

id	Dataset		Clusters			
a	comp vs rec	Source	comp-1	comp-4	rec-2	rec-4
		Target	comp-2	comp-5	rec-1	rec-3
b	comp vs sci	Source	comp-1	comp-2	sci-1	sci-2
		Target	comp-4	comp-5	sci-3	space
c	comp vs talk	Source	comp-1	comp-5	talk-2	talk-4
		Target	comp-2	comp-3	talk-1	talk-3
d	rec vs talk	Source	rec-1	rec-2	talk-1	talk-3
		Target	rec-3	rec-4	talk-2	talk-4
e	sci vs talk	Source	sci-2	sci-3	talk-3	talk-4
		Target	sci-1	sci-4	talk-1	talk-2

Table 2. 6 clusters dataset

id	Dataset		Clusters					
f	comp vs rec vs sci	Source	comp-1	comp-2	rec-1	rec-2	sci-1	sci-2
		Target	comp-3	comp-4	rec-3	rec-4	sci-3	sci-4
g	comp vs rec vs talk	Source	comp-1	comp-2	rec-1	rec-2	talk-1	talk-2
		Target	comp-3	comp-4	rec-3	rec-4	talk-3	talk-4
h	comp vs sci vs talk	Source	comp-1	comp-2	sci-1	sci-2	talk-1	talk-2
		Target	comp-3	comp-4	sci-3	sci-4	talk-3	talk-4
i	rec vs sci vs talk	Source	rec-1	rec-2	sci-1	sci-2	talk-1	talk-2
		Target	rec-3	rec-4	sci-3	sci-4	talk-3	talk-4

3 Evaluations

3.1 Experimental Settings

Datasets. Based on previous work [10], we conducted experiments on 20 Newsgroup data (20NG)[1]. Clustering of these datasets corresponds to document clustering. 20NG contains top 4 categories {comp,rec,sci,talk}, and sub-categories are included under them. As in [10], from the same top category, the source and the target domains are set to different sub-categories. The objective of transfer learning is to improve performance in the target domain (only small amount of data is available) by utilizing large amount of data in source domain, Thus, the amount of data in source domain was set to four times larger than that in the target domain, and 25 documents were sampled from each sub-category in the target domain. This process was repeated and 10 samples were created for each dataset. The utilized datasets are shown in Table 1 and Table 2.

For each sample, we conducted stemming using porter stemmer[2] and MontyTagger[3], removed stop words, and selected 2,000 words with large mutual information [2].

Evaluation Measures. For each dataset, the cluster assignment was evaluated with respect to Normalized Mutual Information (NMI). Let C, $\hat{C}$ stand for the random variables over the true and assigned clusters. NMI is defined as

[1] http://people.csail.mit.edu/~jrennie/20Newsgroups/. 20news-18828 was utilized.
[2] http://www.tartarus.org/~martin/PorterStemmer
[3] http://web.media.mit.edu/~hugo/montytagger

$NMI = \frac{I(\hat{C};C)}{(H(\hat{C})+H(C))/2}$ $(\in [0,1])$ where $H(\cdot)$ is Shannon Entropy, $I(;)$ is Mutual Information. NMI corresponds to the accuracy of cluster assignment. The larger NMI is, the better the result is.

Comparison. We utilized the proposed method on 1) NMF [9], 2) WNMF [14], 3) GNMF [1], and evaluated its effectiveness. In addition, we compared with other transfer leaning methods: 4) SDT [10]C 5) MTrick [15]. In addition, we compared with the previous approach for document clustering with NMF in eq.(3), *i.e.*, by setting the number of clusters k to the number of features q and using argmax for cluster assignment. We assumed that the number of clusters k is specified.

WNMF [9] first converts the data matrix $\mathbf{X}$ by utilizing the weighting scheme in Ncut [13], and applies the standard NMF algorithm on the converted data. GNMF [1] utilizes the first and the third term in eq.(15) as explained in Section 2.6. For the conventional NMF and WNMF, cluster assignment was determined using eq.(3). However, since the results of GNMF using eq.(3) was very poor, skmeans was applied to the constructed representation $\mathbf{V}_t$ by GNMF.

Parameters. Cosine similarity, which is widely utilized in text processing, was utilized as the pairwise similarity measure. Although SDT andMTrick assume label information in the source domain, no label information was utilized in the evaluation. Thus, the regularization parameter for the label information was set to 0 in these methods. Based on [10], the coefficient for the target domain was set to 0.025 in SDT, and kmeans was utilized.

Based on [15], in MTrick, the number of word clusters was set to 50, and the coefficient for the target domain was set to 1.5. Since logistic regression based on the label information cannot be utilized as described in [15], the matrix $\mathbf{G}$ in $\mathbf{X} \simeq \mathbf{FSG}^T$ was initialized using skmeans in our experiments. On the other hand, following [15], $\mathbf{F}$ was initialized using pLSI [7]. The number of neighbors m was set to 10 in GNMF, and λ was set to 100 based on [1].

Evaluation Procedure. For the constructed $\mathbf{V}_t$ in the target domain, kmeans and skmeans were applied to conduct clustering. Since NMF finds out local optimal, the obtained matrices $\mathbf{U}$ and $\mathbf{V}$ depend on the initialization. Thus, we conducted 10 random initialization for the same data matrix. This process is repeated in 10 samples for each dataset[4]. The number of maximum iterations in NMF was set to 30.

3.2 Evaluation on Real-World Datasets

The horizontal axis corresponds to the number of features q, the vertical one to NMI. In the legend, solid lines correspond to NMF, dotted lines to WNMF, and dash lines to GNMF. In addition, +T stands for the results by utilizing the proposed method in Section 2 for each method. Based on our preliminary experiments, ν was set to 0.15 and λ to 1.5 in the following experiments.

[4] The average of 100 runs is reported in each dataset.

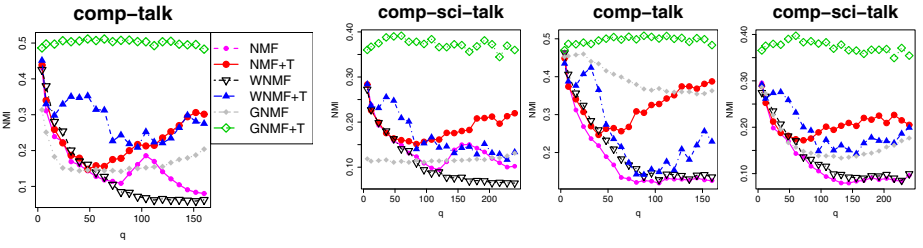


Fig. 3. Effects of # features q (left: kmeans, right: skmeans)

Table 3. Comparison (NMI, with skmeans)

qRatio	Dataset	a	b	c	d	e	f	g	h	i
$k \times 10$	NMF+T	0.228	0.105	0.211	0.159	0.180	0.231	0.250	0.202	0.225
	WNMF+T	0.325	0.123	0.308	0.158	0.187	0.163	0.235	0.291	0.204
	GNMF+T	**0.527**	**0.287**	**0.447**	**0.329**	**0.413**	**0.428**	**0.462**	**0.449**	**0.423**
$k \times 30$	NMF+T	0.310	0.148	0.294	0.216	0.245	0.266	0.301	0.263	0.263
	WNMF+T	0.214	0.109	0.152	0.125	0.161	0.152	0.196	0.207	0.164
	GNMF+T	0.516	0.252	0.413	0.295	0.351	0.385	0.431	0.429	0.349
other transfer	MTrick	0.441	0.283	0.396	0.284	0.339	0.383	0.409	0.396	0.364
	SDT	0.358	0.118	0.210	0.158	0.154	0.160	0.188	0.254	0.179
with max (eq.(3)) ($q = k$)	NMF	0.435	0.249	0.387	0.281	0.331	0.358	0.386	0.377	0.354
	WNMF	0.315	0.215	0.364	0.235	0.273	0.262	0.356	0.326	0.312
	GNMF	0.393	0.161	0.345	0.243	0.247	0.308	0.332	0.326	0.276

Effectiveness of Transfer Learning. It is important to learn sufficient number of features in representation learning, especially in Self-Taught Learning [12]. Thus, we conducted experiments by varying the number of features q and evaluated the effectiveness of the proposed method. As an example, the results for comp vs talk (c) in Table 1 and comp vs sci vs talk (h) in Table 2 are illustrated in Fig. 3 (left graphs are for kmeansC and right ones for skmeans).

In Fig. 3, the performance of the original NMFCWNMF, GNMF decreased as the number of features q increases. On the other hand, by utilizing the proposed transfer learning method, GNMF+T (green dash lines with diamonds) was very robust with respect to the increase in q, and showed much better performance. The performance of NMF+T and WNMF+T decreased at first but later increased as q increased. In addition, as in GNMF, the proposed method showed the improvement on these methods when the number of features is large.

Comparison with other methods. Comparison with other methods is summarized in Table 3. The role of source and target domains in Table 1 and Table 2 were also exchanged, and their average with skmeans is shown in Table 3[5]. Al-

[5] When q is increased, the representation of $\mathbf{V}_t$ becomes high-dimensional. Thus, skmeans showed slightly better performance than kmeans.

though the performance depends on the number of features q, Table 3 shows the cases when q is 10 times larger than k (in Table 3, the row for $k \times 10$) and 30 times larger (the row for $k \times 30$).

From Table 3, we can see that the proposed GNMF+T in eq. (15) outperformed all the other methods. Especially, compared with the original GNMF (in the row for $q = k$), the results show the performance improvement with the proposed method. In GNMF+T and WNMF+T, the results for $k \times 10$ showed better performance. On the other hand the results $k \times 30$ was better in NMF+T since its performance improved as q increases (see Fig. 3). It outperformed SDT, but not MTrick. The performance decreased much faster in WNMF+T w.r.t. q.

In terms of clustering performance, the proposed method with eq.(15) (GNMF+T) showed the best result. On the other hand, since the performance of NMF+T and WNMF+T decreased as the number of features q increased, they could not outperform the conventional NMF and WNMF (with eq. (3) by setting q=k).

3.3 Discussions

The results in Section 3.2 indicate that the proposed transfer learning method is effective, even when many features were utilized. When the local representation $\mathbf{U}$ learned by NMF is utilized for transfer learning, it is important to learn enough "pool" of features as in [12]. Thus, we believe that the proposed method is useful for such situations. The performance of conventional NMF methods drastically decreased as the number of features q increased. On the other hand, via transfer learning, the proposed method *increased* the performance as q increased; especially, it was effective and robust for GNMF+T with eq. (15). However, although the method with eq.(15) outperformed other methods, other methods could not outperform the conventional NMF and WMF (with eq. (3) by setting q=k).

Our transfer hypothesis is the similarity of feature spaces in NMF, and the proposed method conducts transfer learning based on the local representation $\mathbf{U}$. Furthermore, contrary to SDT and MTrick, no "label" information is required to conduct transfer learning in our approach. The performance of SDT was rather low since no label information was utilized in the evaluation. On the other hand, the performance of MTrick strongly depended on the initial values in matrices[6]. It would be possible to improve the performance of NMF-based methods by modifying the initialization of matrices, but this is out of the scope of this paper.

4 Concluding Remarks

We proposed a transfer learning method based on Non-negative Matrix Factorization (NMF). Since NMF learns feature vectors to approximate the given data, the proposed method tries to preserve the feature space which is spanned by the feature vectors in transfer learning. Based on the learned feature vectors

[6] When $\mathbf{G}^T$ is randomly initialized as in NMF, the clustering performance of MTrick was very low. Thus, instead, we utilized skmeans in the evaluation.

in the source domain, a graph structure called topic graph is constructed, and the graph is utilized as a regularization term in the framework of NMF. We show that the proposed regularization term corresponds to maximizing the similarity between topic graphs in both domains, and that the term corresponds to the graph Laplacian [13] of the topic graph. Furthermore, we proposed a learning algorithm with multiplicative update rules, which is guaranteed to converge.

Contrary to other method, no "label" information is required to conduct transfer learning in our approach. The proposed method was evaluated over document clustering problem, and the results indicated that the proposed method improved performance via transfer learning. Especially, the proposed method showed large performance improvement even when many features were utilized.

References

1. Cai, D., He, X., Wu, X., Han, J.: Non-negative matrix factorization on manifold. In: Perner, P. (ed.) ICDM 2008. LNCS (LNAI), vol. 5077, pp. 63–72. Springer, Heidelberg (2008)
2. Cover, T., Thomas, J.: Elements of Information Theory. Wiley, Chichester (2006)
3. Dai, W., Xue, G.-R., Yang, Q., Yu, Y.: Co-clustering based classification for out-of-domain documents. In: Proc. of KDD 2007, pp. 210–219 (2007)
4. Dhillon, J., Modha, D.: Concept decompositions for lage sparse text data using clustering. Machine Learning 42, 143–175 (2001)
5. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: Proc. of KDD 2006, pp. 126–135 (2006)
6. Harville, D.A.: Matrix Algebra From a Statistican's Perspective. Springer, Heidelberg (2008)
7. Hofmann, T.: Probabilistic latent semantic indexing. In: Proc. of SIGIR 1999, pp. 50–57 (1999)
8. Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: Proc. of IJCAI 2003, pp. 561–566 (2003)
9. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Proc. of Neural Information Processing Systems (NIPS), pp. 556–562 (2001)
10. Ling, X., Dai, W., Xue, G., Yang, Q., Yu, Y.: Spectral domain-transfer learning. In: Proc. of KDD 2008, pp. 488–496 (2008)
11. Pan, S.J., Yang, Q.: A survey on transfer learning, pp. 1345–1359 (2009)
12. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.: Self-taught learning:transfer learning from unlabeled data. In: Proc. of ICML 2007, pp. 759–766 (2007)
13. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
14. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proc. of SIGIR 2003, pp. 267–273 (2003)
15. Zhuang, F., Luo, P., Xiaong, H., He, Y., Xiong, Q., Shi, Z.: Exploiting associations between word clusters and document classes for cross-domain text categorization. In: Proc. of ICDM 2010, pp. 13–24 (2010)

Compression and Learning in Linear Regression

Florin Popescu and Daniel Renz

Fraunhofer FIRST, Intelligent Data Analysis - IDA

Abstract. We introduce a linear regression regularization method based on the minimum description length principle, which aims at both sparsification and over-fit avoidance. We begin by building compact prefix free encryption codes for both rational-valued parameters and integer-valued residuals, then build smooth approximations to their code lengths, as to provide an objective function whose minimization provides optimal lossless compression under certain assumptions. We compare the method against the LASSO on simulated datasets proposed by Tibshirani [14], examining generalization and accuracy in sparsity structure recovery.

1 Introduction

1.1 Minimum Description Length.

The Minimum Description Length (MDL) principle is one of many model selection criteria that have been proposed to tackle the general model selection problem. Based on algorithmic information theory [13], it was first formulated by Rissanen in 1978 [10]. In the past decades several theoretical approaches to MDL have evolved [7], while practical implementations remain an open research topic [12], as the computational burden MDL can impose is significant.

Practical MDL uses description methods that are less expressive than universal languages, confined to simplified model classes. It restricts the set of allowed codes in a manner which allows us to find the shortest code length of the data, relative to the set of allowed codes in finite time [12, 13]. So far, three practical MDL schemes have been devised [9]. We used the simplest of them, which is called *two-part MDL*: $\hat{M} = \text{argmin}_{M \in \mathcal{M}} \left(L\left(D|M\right) + L\left(M\right) \right)$, where D is the data, M the model, $\mathcal{M}$ the model class (the set of allowed models) and L the code length function. If we knew the probability distributions $p(M), p(D|M)$, we could use Bayesian or ML estimation and use appropriate asymptotically compact codes, i.e. Shannon-Fano coding, to generate the code books for $L(D|M)$ and $L(M)$ (as it is done in Wallace's Minimum Message Length). As these are not known *a priori*, we must make more general assumptions.

1.2 Regularized Linear Regression vs. Compressive Linear Regression

In linear regression, we assume data is organized in an observation matrix $D \in \mathbb{R}^{J \times \mathrm{N}}$ where N is the number of observations and J is the number of

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 270–279, 2011.

features. This matrix may be expanded into a matrix $X(D)$ in two ways: by adding functions of groups of observations (therefore increasing N), as in kernel expansion, or adding rows to D, which are functions of the original features:

$$\boldsymbol{y} = X(D)\boldsymbol{\theta} + \boldsymbol{e},\ \boldsymbol{y} \in \mathbb{R}^N, \boldsymbol{\theta} \in \mathbb{R}^K, \tag{1}$$

where $K = J+p$, $\boldsymbol{\theta}$ is the vector of linear regression parameters, $\boldsymbol{e}$ is the residual and $\boldsymbol{y}$ is the vector we call herein the target or output observation. Depending on our prior knowledge of the probability density function $p(\boldsymbol{e})$ (i.e. of measurement errors) there are various ways in which an approximate solution $\hat{\boldsymbol{\theta}}$ may be sought, by minimizing the appropriate loss function (or log-likelihood). One of the most common "non-uniformative" assumptions, consistent with a maximal entropy distribution for finite variance i.i.d sampling, is that the errors are Gaussian and the corresponding loss function to be minimized is provided by the 2-norm

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left(\|\boldsymbol{y} - X(D)\boldsymbol{\theta}\|_2 + \gamma \|\boldsymbol{\theta}\|_r \right) \tag{2}$$

Regularization ($\gamma \neq 0$) of ℓ_2 regression is helpful in avoiding overfit, as well as in providing a unique rather than infinite solution space in the underdetermined case. Various regularization norms have been proposed, the most commonly used being ridge regression and LASSO [14]. The Tikhonov regularization [15] proposes a generalized 2-norm, but commonly the straightforward ℓ_2 norm is used (ridge regression). The LASSO proposes the ℓ_1 norm, offering the additional advantage of a tendency to sparsify $\boldsymbol{\theta}$. Both approaches can be seen, from a Bayesian perspective, to assume a Gaussian error distribution $p(\boldsymbol{e})$ and independent zero-mean priors with uniform variances over components of $\boldsymbol{\theta}$, of Gaussian form for ridge regression and double-exponential for the LASSO. Our objective was to provide for a scale invariant objective function which allows for meaningful Bayesian interpretation and does not require hyper-parameter tuning. The MDL formulation of regularized regression requires, first of all, the weak assumption that the target has finite and uniform measurement precision or quantization width $\delta(\boldsymbol{y})$ (i.e. $\boldsymbol{y}/\delta(\boldsymbol{y}) \in \mathbb{Z}^N$). We seek

$$\min_{\boldsymbol{\theta}_\#} \left(L\left(\frac{\boldsymbol{y}}{\delta(\boldsymbol{y})} - \left\lfloor \frac{X(D)\boldsymbol{\theta}_\#}{\delta(\boldsymbol{y})} \right\rfloor \right) + L\left(\boldsymbol{\theta}_\#\right) \right), \tag{3}$$

where $\boldsymbol{\theta}_\# \in \mathbb{Q}^K$. The difference between equations (2) and (3) is that we are optimizing over rational-valued vectors $\boldsymbol{\theta}_\#$, rather than real-valued ones, as both the parameter prior and the measurement error distributions are now discrete probabilities rather than probability density functions. Clearly, a direct optimization over rational numbers would be very difficult. This is precisely why we develop codes, approximations and bounds to make MDL optimization possible for the model class of linear regression models with nonlinear feature products. We shall call this method *Compressive Linear Regression (CLR)*.

2 Methods

The approach we will follow is to compute smooth approximations of description lengths of regression parameters and residuals for the purpose of optimization: compact codes for rationals and random integer sequences must be revisited.

2.1 Coding of Integers

Rissanen's universal prefix-free code for integers [11] provides a Bayesian "non-uniformative" prior $2^{-L(i)}$, a principle which can be extended from the integer set to rational numbers as well, which we will propose in this paper. A universal code is a prefix-free code which has an expected code length over all integers that is invariant, within a constant, no matter what the prior probability of the integers is (assuming that it monotonically decreases). However, Rissanen's code is not very compact as defined by the Kraft inequality: $\Theta = \sum_{d \in \mathcal{D}} 2^{-L(d)} \leq 1$. Rissanen's code is not convex, being concave at $x_1 = 2, x_2 = 2^2 = 4, x_3 = 2^4 = 16, x_4 = 2^{16}, ...$, which can cause problems for some optimization algorithms. Universal codes are the basic building block for the other codes described, therefore we chose to build a smooth, compact and convex universal code U_n for non-negative integers and for signed integers as well via the ordering 0,-1,1,-2,2,-3 etc. We call the code for the signed integers U. Our code combines two different coding schemes: For small numbers, a coding scheme F similar to Fibonacci coding [1] switches at a certain point to a coding scheme E producing code words with the same lengths as Elias-Delta coding [5] for large numbers.

2.2 Coding of Rationals within a Real Line Interval

Any universal integer code can be used to construct a prefix-free code for rational numbers of specified precision. However, there is no obvious ordering scheme for rationals in terms of code lengths. Li and Vitányi offer a method to map rationals into the unit interval $[0, 1] \in \mathbb{R}$ using the idea of cylinder sets [16]. We implement a variant of this code and name it α-code:

$$\alpha : \mathbb{R}^2 \rightarrow \{0,1\}^* \text{ , } \alpha^{-1}\text{: } \{0,1\}^* \rightarrow \mathbb{Q} \text{ , s.t.}$$
$$\theta_{\#} = \alpha^{-1}\left(\alpha(\theta, \delta)\right) \in (\theta - \delta, \theta - \delta)) \tag{4}$$

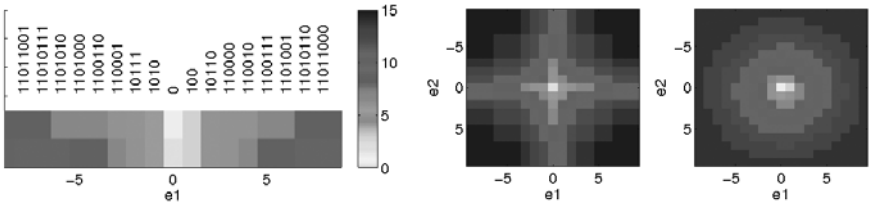


Fig. 1. Integer codelengths. Left plot, bottom row: Risannen codelengths. Middle row: codelengths for the F code. Top row: codewords for F. Middle: Codelengths for independent elements of $e = (e_1, e_2...)^T$ using F. Right: Spherical coding of e using F.

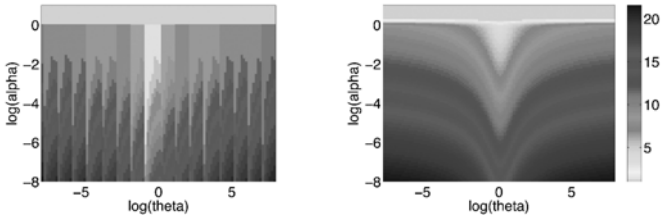


Fig. 2. Left: α code for rational numbers. Code word lengths increase with $|\theta|$ and θ/δ. Right: Smooth approximation function for the α coding function. $\bar{\alpha} = c_0 \cdot \log\left(\tau(\theta) \cdot \left(\operatorname{erf}\left(10 \cdot (1 - \delta/\theta) + 1\right) \cdot |\theta| + \delta\right) - \log \delta\right) + 1$. $\tau(\theta) = \left|c_1 \cdot \left(\log\left(\log\left(\theta^2 + c_2\right)\right)\right) - 1\right|$. `erf` is the error function and c_0, c_1 and c_2 are numerically fitted constants of O(1).

where $\alpha^{-1}(\alpha(\theta, \delta))$ is the rational number coded by the binary $\alpha(\theta, \delta)$. Using the code α, we can recover a rational number close to the *real* number θ to a precision of at least δ. We define $\alpha = \alpha(\theta, \delta) = U(\theta_\delta)U(\theta_{\log})$, where θ is the rational number to be encoded, δ the precision to which it should be encoded, $\theta_\delta = \lceil \theta/\delta \rceil$ and $\theta_{\log} = \lceil \log |\theta| \rceil - \lceil 0.5 \log \log \theta_\delta \rceil$. The decimal value of the code word $\alpha(\theta, \delta)$ is $\theta_\# = \theta_\delta \cdot 2^{\lceil \log \theta_\delta \rceil + \theta_{\log}}$.

As α is a prefix-free code, the code length for a vector of rationals is simply the length of the concatenation of the codes of its elements: $\alpha(\boldsymbol{\theta}, \boldsymbol{\delta}) = \sum_i \alpha(\theta_i, \delta_i)$. We shall use α to store the linear regression parameters. Through numerical tests, we were able to find a smooth function $\bar{\alpha}$ that approximates our α-code lengths, for which we calculated an approximation mean error of 0.8 bit by sampling 10^5 evenly spaced points for $\theta \in [2^{-8}; 2^8]$ and a *relative precision* $\theta/\delta \in [2^{-8}; 2^0]$. The α code length and its smooth approximation is shown in Figure 2.

2.3 Spherical Coding of a Signed Integer Sequence

We seek a means of coding the part of the data that cannot be explained by the model, i.e. the residuals $\boldsymbol{e}$ (a sequence of signed integers - see Fig. 1). Since only model fitting (compression) can determine that the sequence is not random, we shall build a code assuming that each element is random (incompressible), independent of each other, and uniform (equivalent to an i.i.d sample). In the case of a physical measurement, we assume the errors are $\delta(X)$ width quantizations of a stationary ergodic continuous stochastic process. We assume that the likelihood of a sample $\boldsymbol{e}$ decreases with its 2-norm (i.e. is "spherical"), which as we will show numerically, is equivalent to assuming a Gaussian error distribution.

The corresponding coding problem becomes one of counting the maximal expected number of hypercubes with side lengths $\delta(X)$ that intersect a hypersphere of N dimensions ordered spirally outwards from the origin (see Fig. 1).

$$H : \mathbb{N} \leftrightarrow \mathbb{Z}^N, \ s.t. \left\|H(i)\right\|_2 \leq \left\|H(i+1)\right\|_2 \ \forall i \in \mathbb{N} \tag{5}$$

The spiral counting scheme $H(i)$ guarantees that the 2-norm is monotonically increasing and therefore, since the universal description length of an integer is

monotonically increasing with the integer, it follows that under this counting scheme the description length of an integer vector under H is monotonically increasing with its 2-norm. This leads us to state that:

$$\begin{aligned} \|\boldsymbol{e}/\delta(X)\|_2 \leq r \;\; &\Rightarrow \;\; K(\boldsymbol{e}) \leq h(r^2, N) \\ h(r^2, N) &\lessapprox \left|U\left(\max\left(V_S^N(r+1), 1\right)\right)\right| \end{aligned} \tag{6}$$

where $V_S^N(r)$ denotes the volume of the hyper-sphere of radius r in $\mathbb{R}^N$. If we know that the 2-norm of an integer vector $\boldsymbol{e}$ is bounded by r, then its description length is less than the universal code length of the number hyper-cubes contained by the sphere of radius $r+1$. We can establish a connection between h and the Shannon entropy $h_{sh}(x)$. It is the shortest average coding length for a random variable x of dimension N. For gaussian data x with variance σ^2,

$$h_{sh}(x) = N/2 \cdot \log(2\pi e) + N \cdot \log(\sigma) \tag{7}$$

To be able to compare h_{sh} and h, we have to consider the case of finite data again: We cannot simply assume that σ is known, but rather we have to actually encode it and consider its coding length. We do so using the α-code. The minimum number of bits that we need to store σ can be found by minimizing over the precision $\Delta\sigma$. We call the resulting *applied* Shannon coding length h_{ap}:

$$h_{ap}(\sigma, \Delta\sigma) = N/2 \cdot \log(2\pi e) + \min_{\Delta\sigma}(N \cdot \log(\sigma + \Delta\sigma) + \alpha(\sigma, \Delta\sigma)) \tag{8}$$

Figure 3 shows that the applied Shannon coding length is approximated by the spherical code length function for large N. Shannon-Fano or Huffman code lengths/entropies for a given distribution are only valid asymptotically, i.e. N very large and only if the standard deviation is known.

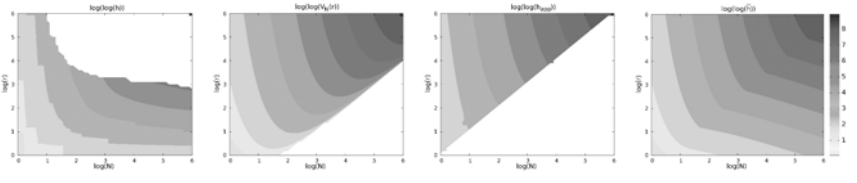


Fig. 3. Spherical counting scheme and approximations: A) surface plot of the exact count h. White indicates the region in which computation time of h was prohibitive. B) volume of an N-dimensional sphere of radius r, $V_S^N(r)$. In the lower right region this approximation diverges from the exact count h, because here the total volume of all hypercubes intersecting the sphere is much larger than the volume of the actual sphere. C) Shannon message length for a gaussian source with distribution $\mathcal{N}(0, r)$. This approximation is consistent with h within order of 1, except in the white region. D) the approximation function $\bar{h}(s_M^2(\boldsymbol{e})) = \log\left(\frac{\pi^{\frac{N}{2}}}{\Gamma\left(\frac{N}{2}+1\right)}\right) + \frac{N}{2}\log\left(\frac{s_M^2(\boldsymbol{\theta}_{\#}, X)}{\delta(X)^2}\right)$.

2.4 MDL for Linear Regression

The complete (loss-less) code length approximation λ_M of the data to be minimized is the sum of two terms: the length of the code for the parameters: the rational code length $\bar{\alpha}(\cdot,\cdot)$ of a rational valued vector and the spherical code length $\bar{h}(\cdot,\cdot)$ of the vector of residuals bounded by the radius $s_M^2(\cdot,\cdot)$:

$$\lambda_M(\boldsymbol{\theta},\boldsymbol{\delta},X) = \bar{\alpha}(\boldsymbol{\theta},\boldsymbol{\delta}) + \bar{h}(s_M^2(\boldsymbol{\theta}_{\#},X),N), \tag{9}$$

where $\boldsymbol{\theta}_{\#} = \alpha^{-1}(\alpha(\boldsymbol{\theta},\boldsymbol{\delta}))$. Bars on α and h denote smooth approximations. Note that this equation requires the function $s_M(\cdot)$, which relates the size of the residual 2-norm to the parameters and their storage precision, for some model M which specifies how the parameters encode $\boldsymbol{y}$. This principle can be applied to any type of regression fit: we shall do so for linear regression.

Finding an analytical function which is a close approximation of $\bar{h}$ is a difficult mathematical problem. For combinations of large enough r (> 5) and low enough N ($< 10^6$) (see previous section) the approximation based on the unit sphere volume (and shown in Figure 2) is O(1) accurate. The α code has a rather nice property: the numerical value of the decoding $\alpha^{-1}(\alpha(\theta_i,\delta_i)) = \theta_i + \delta(\theta_i)$ is uniformly distributed on the interval $[\theta_i - \delta_i, \theta_i + \delta_i]$ with $p_i(\delta) = (2\delta_i)^{-1}$, over all possible pairs of θ and δ: as Monte-Carlo simulations have confirmed. This allows us to take the expected value of the increase in the bound of the norm of the residual as the effect of random, independent perturbations $\delta(\theta_i)$ of each θ_i with the uniform probability $p_i(\delta)$:

$$s^2(\alpha^{-1}(\alpha(\boldsymbol{\theta},\boldsymbol{\delta})),X) = (\boldsymbol{y} - X\cdot(\boldsymbol{\theta}+\boldsymbol{\delta}(\boldsymbol{\theta})))^T\cdot(\boldsymbol{y} - X\cdot(\boldsymbol{\theta}+\boldsymbol{\delta}(\boldsymbol{\theta})))$$

We choose $\boldsymbol{\theta} = \left(X^TX\right)^{-1}X^Ty$ to minimize the 2-norm of the residual $\boldsymbol{e}^T\boldsymbol{e}$, (or any exact solution in the underdetermined case) so the gradient is with respect to $\boldsymbol{\theta}$ zero, meaning that any perturbation from $\boldsymbol{\theta}$ results in strictly quadratic growth: $\boldsymbol{e}^T\boldsymbol{e} + \boldsymbol{\delta}(\boldsymbol{\theta})^TX^TX\boldsymbol{\delta}(\boldsymbol{\theta}) = \boldsymbol{e}^T\boldsymbol{e} + \boldsymbol{\delta}(\boldsymbol{\theta})^T\Sigma_X\boldsymbol{\delta}(\boldsymbol{\theta})$, where Σ_X is the covariance of X. Taking the expected value over possible perturbations:

$$E\left[\delta(\theta_i)^T(\Sigma_X)_{i,i}\delta(\theta_i)\right] = \int_{-\delta_i}^{\delta_i} p(\delta(\theta_i))(\Sigma_X)_{i,i}\delta(\theta_i)^2 d\delta(\theta_i) = \frac{1}{3}(\Sigma_X)_{i,i}\delta_i^2$$

$$E\left[\delta(\theta_i)^T(\Sigma_X)_{i,j}\delta(\theta_j)\right]_{i\neq j} = \int_{-\delta_j}^{\delta_j}\int_{-\delta_i}^{\delta_i}(\Sigma_X)_{i,j}\delta(\theta_i)\delta(\theta_j)d\delta(\theta_i)d\delta(\theta_j) = 0$$

$$E\left[\delta(\boldsymbol{\theta})^T\Sigma_X\delta(\boldsymbol{\theta})\right] = E\left[\sum_{i,j}\delta(\theta_i)^T(\Sigma_X)_{i,j}\delta(\theta_j)\right] = \frac{1}{3}\sum_i(\Sigma_X)_{i,i}\delta_i^2$$

A differentiable CLR objective function for linear regression can now be written.

$$\min_{\boldsymbol{\theta}}\left(E_{\delta(\theta)}\left[\lambda_M(\boldsymbol{\theta},\boldsymbol{\delta},X)\right]\right) = \min_{\boldsymbol{\theta}}\left(\bar{\alpha}(\boldsymbol{\theta},\boldsymbol{\delta}) + E_{\delta(\theta)}\bar{h}(s_M^2 + \delta(X)^T\Sigma_{X,M}\delta(X),N)\right)$$

$$\approx \min_{\boldsymbol{\theta}}\left(\sum_i\bar{\alpha}(\theta_i,\delta_i) + \bar{h}(s_M^2 + \frac{1}{3}\sum_i(\Sigma_X)_{i,i}\delta_i^2,N)\right)$$

The approximation $E[h(a+x)] = E[h(a)+h'(a)x+...] = h(a)+h'(a)E[x]+... \cong h(a + E[x])$ is valid if $h(a)$ is locally linear - being $O(\log(a))$ that assumption is reasonable. The resulting objective function is smooth in optimization parameters ($2K$ such parameters: $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$) but is non-convex. A useful unbiased heuristic we have found was to use simplex optimization (code was written in Matlab, Mathworks, Inc.) with starting value for $\boldsymbol{\theta}$ at the usual 2-norm solution and $\delta_i = |\theta_i|/2$. After each downhill optimization, all parameters for which $\delta_i > |\theta_i|$ were discarded and the entire procedure repeated (with reduced X) until no further parameters were thus 'culled'.

3 Results

Simulated data allows us to gauge the relative ability of CLR to accurately recover the sparsity structure in the case in which it is known. We produced the same synthetic data sets as did Tibshirani in his seminal LASSO paper [14]. For the first 3 example sets (TIB1, 2 and 3), we simulated 50 instances consisting of N=20 observations from the model $y = \boldsymbol{\theta}^T\boldsymbol{x} + \sigma\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is standard Gaussian. The correlation between x_i and x_j was $\rho^{|i-j|}$ with $\rho = 0.5$. The dataset parameters for TIB1 were $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)^T$ and $\sigma = 3$, for TIB2 $\beta_j = 0.85, \forall j$ and $\sigma = 3$ and for TIB3 $\boldsymbol{\beta} = (5, 0, 0, 0, 0, 0, 0, 0)$ and $\sigma = 2$. In a fourth example TIB4, we simulated 50 data sets consisting of 100 observations, with $x_{ij} = z_{ij} + z_i$ where z_{ij} and z_i are independent Gaussian variates, $\beta = (0, 0, ..., 0, 2, 2, ..., 2, 0, 0, ..., 0, 2, 2..., 2)$ there being

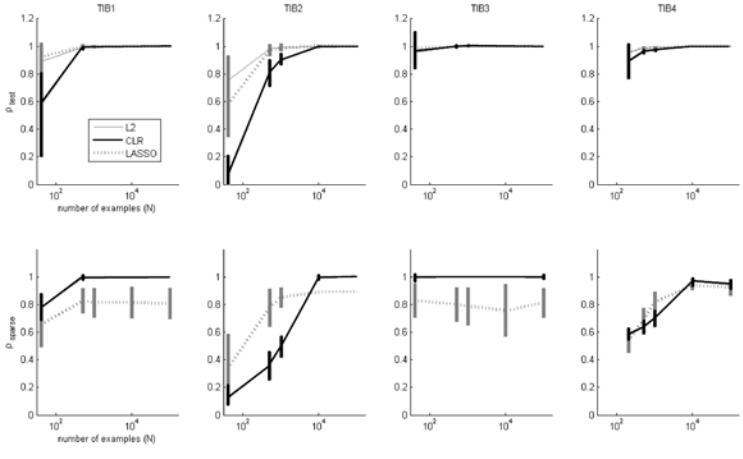


Fig. 4. Results for the datasets TIB1-4 described in [14] as functions of the number of examples. Top row: Mean Spearman correlation coefficient of prediction to test data. Bottom row: Mean Spearman's correlation of sparsity structure (0 if θ_i=0, 1 otherwise).

10 repeats in each block, and $y = \boldsymbol{\beta}^T \boldsymbol{x} + 15\epsilon$, where ϵ was standard normal. In addition to these datasets, which are exactly as in Tibshirani's original paper, the same stochastic processes were simulated for 500, 1000, 5000 and 10000 data points in order to gage sparsity structure recovery. In [14] LASSO performed favorably compared to other methods such as ridge regression and therefore only LASSO was used for comparison (see Figure 4). Our LASSO implementation used bootstrapping (n=128) for LASSO statistic estimation and 10-fold CV for model selection. We used a LASSO implementation (www.mathworks.com/matlabcentral/fileexchange) written by M. Dunham and which is an implementation of BOLASSO [3] with LARS as the workhorse LASSO routine. Other LASSO implementations, namely largest common region model selection and non-bootstrapped CV LASSO performed worse and are not included. The average number of parameters returned for each set, with target value being {2,8,1,20} was, for LASSO, {5.06,3.63,2.76, 16.4} and for CLR {4.37, 3.04, 3.31, 5.76} (the bias value is not included). CLR used simplex minimization with random multistart (no. of threads on the order of number of parameters). We tested generalization performance on a dataset from the UCI ML repository (http://www.ics.uci.edu/˜mlearn/MLRepository.html). We used random splits with equal training / test set sizes, averaged over 350 runs. For triazines (N=186, K=61), the test error correlations were 0.2984 (REG), 0.2890 (CLR) and 0.3519 (LASSO), with sparsity ratios of 0.105 (CLR) and 0.251 (LASSO).

4 Conclusion

The numerical results of CLR vs. LASSO, can be summarized briefly: while both methods sparsify, CLR showed underfit for low numbers of examples relative to LASSO, but was more accurate in recovering sparsity structure ('sparsistency') with increasing number of examples. For moderate number of examples, BOTH LASSO and CLR were comparable in both test error and sparsistency, although they provided different answers for individual examples. Explanations may be found in the nature of the two approaches, The LASSO is not, as sometimes mistakenly assumed, a convex method. Cross-validated objective functions are in fact highly nonconvex in one crucial parameter: the hyperparameter λ in Eqn. (2). With a fixed λ LASSO is both convex and asymptotically consistent: but rarely is it fixed *a priori*. While progress has recently been made in providing algorithms such as the LASSO with a quicker and more robust choice of λ [2], the BOLASSO method, originally introduced by Tibshirani, still chooses the hyperparameter with the lowest expected generalization error thus placing sparsistency as secondary priority. Had we introduced a tunable hyperparameter in CLR, Eqn. (9) the results would have changed favorably in terms of test error at all data lengths. However, this would be contrary to the MDL principle and our aim to provide a consistent and unambiguous principle for learning in regression.

Feature selection is decidedly NP hard (7), and while tractability and speed are important, there is no universal principle which compels belief in low 1-norms of regressors, or in the Bayesian prior they imply. Since all features are

dimensional, a 1-norm of the regressor is scale variant, a problem which feature normalization would only partially resolve, since feature scale is not guaranteed to generalize. Similar ambiguities apply to other more explicit Bayesian approaches [4] data-adaptive methods [6] and mixed Tikhonov/LASSO approaches. Instead, our approach relies on a principled, scale invariant but non-convex, cost function: the approximated description length. The coding methods can be used to actually compress data, but that is not the main point. The feature selection method we have employed was both forward selection (variable ranking) and backward elimination [8], by which parameters whose approximation interval includes 0 are eliminated. CLR's behaviour in regards to sparsification is different to BOLASSO: at low numbers of examples or small amount information in the predicted variable (e.g. binary targets) the increase in parameter storage is not balanced by residual storage requirements, but as N increases the situation reverses. The running time was similar between LASSO and CLR (for large number of points CLR was faster, otherwise not): (BO)LASSO spends computational resources on CV/bootstraping and CLR on nonconvex optimization: a direct comparison would depend on implementation efficiency and platform.

In terms of generalization error, the simulated data sets were more appropriate for sparsistency evaluation due to the limited number of features relative to examples which precludes the possibility of overfit (in fact, regular L2 regression performed best for simulated data even though it did not sparsify). A convincing test of over-fit avoidance would have to be performed on a large variety of real datasets, in which the learning problem would presumably be nonlinear and in which the number of features is large compared to the number of examples, such as TRIAZINES, for which CLR works best (while using fewest features). In other UCI datasets with non-binary targets regularization was not necessary.

Our work is a beginning which points to the possibility of fully automatic machine learning. The model types upon which MDL framework and the regression regularization proposed apply are not restricted to linear regression or Gaussian errors (application to other model types is a matter of adapting codes and for parameters and residuals), but the optimization problem nonlinear models entail can be even more challenging. Future work is needed to provide nearly optimal solutions to compression problems in affordable polynomial time, paralleling the abundant work on the Traveling Salesman Problem, in which Euclidean distance is an analogy for description length of data.

References

[1] Apostolico, A., Fraenkel, A.S.: Robust transmisson of unbounded strings using fibonacci representations. IEEE Transactions on Information Theory, IT-33(2), 238–245 (1987)

[2] Arlot, S., Bach, F.: Data-driven calibration of linear estimators with minimal penalties. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22. MIT Press, Cambridge (2009)

[3] Bach, F.R.: Bolasso: model consistent lasso estimation through the bootstrap. In: Proceedings of the 25th International Conference on Machine Learning, vol. 307, pp. 33–40 (2008)
[4] Cawley, G.C., Talbot, N.L.C., Girolami, M.: Sparse multinomial logistic regression via bayesian l1 regularisation. In: Advances in Neural Information Processing Systems 19. MIT Press, Cambridge (2007)
[5] Elias, P.: Universal codeword sets and representations of the integers. IEEE Transactions on Information Theory, IT-21(2), 194–203 (1975)
[6] Figueiredo, M.: Adaptive sparseness for supervised learning. IEEE Transations on Pattern Analysis and Machine Intelligence 9(25), 1150–1159 (2003)
[7] Grunwald, P.: Model selection based on minimum description length. Journal of Mathematical Psychology 44(1), 133–152 (2000)
[8] Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction, Foundations and Applications. Physica-Verlag, Springer (2006)
[9] Lanterman, A.D.: Schwarz, wallace and rissanen: Intertwining themes in theories of model selection. International Statistical Review 69(2), 185–212 (2001)
[10] Rissanen, J.: Modeling by shortest data description. Automatica 14, 465–471 (1978)
[11] Rissanen, J.: Universal coding, information, prediction, and estimation. IEEE Transactions on Information Theory, IT-30(4), 629–636 (1984)
[12] Schmidhuber, J.: The speed prior: A new simplicity measure yielding near-optimal computable predictions. In: Kivinen, J., Sloan, R.H. (eds.) COLT 2002. LNCS (LNAI), vol. 2375, pp. 216–228. Springer, Heidelberg (2002)
[13] Solomonoff, R.: A formal theory of inductive inference, part i and ii. Information and Control, Part I/II 7(1/2), 1–22, 224–254 (1964)
[14] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society B (Methodological) 58(1), 267–288 (1996)
[15] Tikhonov, A., Arsenin, V.: Solution for Ill-Posed Problems. Wiley, New york (1977)
[16] Vitanyi, P., Li, M.: Minimum description length induction, bayesianism, and kolmogorov complexity. IEEE Transactions on Information Theory 46(2), 446–464 (2000)

The Impact of Triangular Inequality Violations on Medoid-Based Clustering

Saaid Baraty, Dan A. Simovici, and Catalin Zara

University of Massachusetts Boston
100 Morrissey Blvd., Boston 02125, USA
{sbaraty,dsim}@cs.umb.edu,
czara@math.umb.edu

Abstract. We evaluate the extent to which a dissimilarity space differs from a metric space by introducing the notion of metric point and metricity in a dissimilarity space. The effect of triangular inequality violations on medoid-based clustering of objects in a dissimilarity space is examined and the notion of rectifier is introduced to transform a dissimilarity space into a metric space.

Keywords: dissimilarity, metric, triangular inequality, medoid, clustering.

1 Introduction

Clustering is the process of partitioning sets of objects into mutually exclusive subsets (clusters), so that objects in one cluster are similar to each other and dissimilar to members of other clusters.

The input data of a clustering technique is the dissimilarity measure between objects. Often, clustering algorithms are applied to dissimilarities that violate the usual triangular inequality (TI) and therefore, fail to be metrics. As we shall see, this compromises the quality of the resulting clustering.

The role of the triangular inequality in designing efficient clustering algorithms has been noted in [1], where it is used to accelerate significantly the k-means algorithm, and in [3], where it is used to improve the efficiency of searching the neighborhood space in the TI-DBSCAN variant of DBSCAN. Another area where violations of the triangular inequality are relevant is the estimation of delays between Internet hosts without direct measurements [4,5]. These violations, caused by routing policies or path inflation impact the accuracy of Internet coordinate systems.

The role of compliance with the triangular inequality in improving the performance of vector quantization has been observed in [7]. Finally, the triangular inequality plays a fundamental role in the anchors hierarchy, a data structure that allows fast data localization and generates an efficient algorithm for data clustering [6].

If a triplet $\{x, y, z\}$ violates the triangular inequality, e.g., we have $d(x, y) > d(x, z) + d(z, y)$, which means that it is possible to have two objects x, y that

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 280–289, 2011.

are similar to a third object z, yet very dissimilar to each other. If x and y are placed in a cluster C whose centroid is z (because of the similarity of x and y with z), the cohesion of C may be seriously impacted by the large dissimilarity between x and y.

The effect of TI violations is studied in the context of medoid-based algorithms. We experiment with PAM (Partition Around Medoids), as described in [2]. This algorithm consists of two phases. In the first phase, BUILD, a collection of k objects (where k is the prescribed number of clusters) called *medoids* that are centrally located in clusters are selected. In the second phase, SWAP, the algorithm tries to improve the quality of the clustering by exchanging medoids with non-medoid objects. We selected PAM over k-means, since it works with a dissimilarity matrix without having the actual coordinates of the points. This allows us to generate dissimilarities in our experiments without having the actual objects at hand.

In Section 2 we introduce dissimilarity spaces and evaluate the extent a dissimilarity is distinct from a metric by using the number of TI violations of a dissimilarity space. The notion of rectifier is introduced in Section 3 as a device for transforming a dissimilarity into a metric such that the relative order of object dissimilarities is preserved. A specific measure for quantifying the impact of TI violations of a dissimilarity on clustering is discussed in Section 4. A measure of quality or coherence of clusters is the topic of Section 5. Finally, we present the results of our experiments in Section 6.

2 Dissimilarity Spaces and Metricity

A *dissimilarity space* is a pair $\mathcal{S} = (S, d)$, where S is a set and $d : S \times S \longrightarrow \mathbb{R}_{\geq 0}$ is a function such that $d(x, x) = 0$ and $d(x, y) = d(y, x)$ for all $x, y \in S$.

If $d(x, y) = 0$ implies $x = y$, then we say that d is a *definite dissimilarity*. If $T \subseteq S$, then the pair (T, d_T) is a *subspace* of (S, d), where d_T is the restriction of d to $T \times T$. To simplify notations we refer to the subspace (T, d_T) simply as T and we denote the restriction d_T by d.

If a dissimilarity satisfies the triangular inequality $d(x, y) \leq d(x, z) + d(z, y)$ for all $x, y, z \in S$, then we say that d is a *semi-metric*. A definite semi-metric is said to be a *metric* and the pair (S, d) is referred to as a *metric space*.

All dissimilarity spaces considered are finite and all dissimilarities are definite. The *range* of the dissimilarity d of a dissimilarity space $\mathcal{S} = (S, d)$ is the finite set $R(d) = \{d(x, y) \mid x \in S, y \in S\}$. Clustering is often applied to dissimilarity spaces rather than to metric spaces. As mentioned in the introduction, we aim to analyze the impact on the quality of the clustering when using dissimilarities rather than metrics.

Let (S, d) be a dissimilarity space and let $z \in S$. The set of metric pairs in z is $M(z) = \{(x, y) \in (S \times S) \mid x, y, z$ are pairwise distinct and $d(x, y) \leq d(x, z) + d(z, y)\}$. The *metricity of* z is the number $\mu(z) = \frac{|M(z)|}{(|S|-1)(|S|-2)}$. An object z is *metric* if $\mu(z) = 1$ and is *anti-metric* if $\mu(z) = 0$.

There exists dissimilarity spaces without any metric point. Indeed, consider $\mathcal{S}_0 = (\{x_1, \ldots, x_n\}, d)$, where $n \geq 4$ and $d(x_i, x_j) = 0$ if $i = j$, $d(x_i, x_j) = 1$ if $|i - j| \in \{1, n-1\}$, and $d(x_i, x_j) = 3$, otherwise. Then, every point x_i is anti-metric because we have $d(x_{i-1}, x_i) = d(x_i, x_{i+1}) = 1$ and $d(x_{i-1}, x_{i+1}) = 3$ (here $x_{n+1} = x_1$ and $x_0 = x_n$).

On the other hand, we can construct dissimilarity spaces with a prescribed proportion of metric points. Consider the set $S_{pq} = \{x_1, \ldots, x_p, y_1, \ldots, y_q\}$, where $q \geq 2$ and the dissimilarity

$$d_{pq}(u, v) = \begin{cases} 0 & \text{if } u = v, \\ a & \text{if } u \neq v \text{ and } u, v \in \{x_1, \ldots, x_p\} \\ b & \text{if } u \neq v \text{ and } u, v \in \{y_1, \ldots, y_q\} \\ c & \text{if } u \in \{x_1, \ldots, x_p\}, v \in \{y_1, \ldots, y_q\}, \end{cases}$$

where $u, v \in S_{pq}$. If $b > 2c \geq a$, then every x_i is non-metric and every y_k is metric. Indeed, any x_i is non-metric because $b = d_{pq}(y_j, y_h) > d_{pq}(y_j, x_i) + d_{pq}(x_i, y_h) = 2c$ if $j \neq h$. In the same time, every y_k is metric because for any choice of u and v we have $d(u, v) \leq d(u, y_k) + d(y_k, v)$, as it can be easily seen.

A *triangle* in a dissimilarity space (S, d) is a subset T of S such that $|T| = 3$. A triangle $T = \{x_i, x_j, x_k\}$ is said to be *metric* if $d(x_{p_1}, x_{p_2}) \leq d(x_{p_1}, x_{p_3}) + d(x_{p_3}, x_{p_2})$ for every permutation (p_1, p_2, p_3) of the set $\{i, j, k\}$. Thus, a triangle $\{x_i, x_j, x_k\}$ is metric if the subspace $\{x_i, x_j, x_k\}$ is metric. The collection of non-metric triangles of the dissimilarity space $\mathcal{S}$ is denoted by $\mathsf{NMT}(\mathcal{S})$.

Observe that for every triangle T of a dissimilarity space (S, d) there is at most one TI violation. Indeed, suppose that $T = \{x, y, z\}$ is a triangle and there are two violations of the TI involving the elements of T, say $d(x, y) > d(x, z) + d(z, y)$ and $d(y, z) > d(y, x) + d(x, z)$. Adding these inequalities and taking into account the symmetry of d we obtain $d(x, z) < 0$, which is a contradiction. A non-metric triangle $\{x, y, z\}$ such that $d(x, z) > d(x, y) + d(y, z)$ is denoted by $T_{(y,\{x,z\})}$.

Theorem 1. *Let (S, d) be a dissimilarity space such that $|S| = n$. The number of TI violations has a tight upper bound of $\binom{n}{3}$.*

Proof: For a collection of n distinct points we have $\binom{n}{3}$ distinct triangles and, by the observation that precedes this theorem, there is at most one TI violation associated with each triangle which establishes the upper bound. To prove that the upper bound is tight, we need to show that there exists a dissimilarity d such that the number of TI violations is exactly $\binom{n}{3}$. That is, each distinct triangle has one TI violation. The dissimilarity d is constructed inductively.

For $n = 3$, $S = \{x, y, z\}$ and we choose $d(x, y), d(x, z)$ and $d(y, z)$ such that there is a TI violation. For example, this can be achieved by defining $d(y, z) = d(x, y) + d(x, z) + 1$.

Let S be a collection of n points with $\binom{n}{3}$ TI violations. Let $S' = S \cup \{u\}$ such that $u \notin S$ and define $d(u, x) = d(x, u) = \frac{\min_{(y,z \in S, y \neq z)} d(y,z)}{2+\epsilon}$ for every $x \in S$, where $\epsilon > 0$. For each newly added triangle $\{u, y, z\}$ we have $d(y, z) > d(y, u) + d(u, z)$, so the number of TI violations in (S', d) is $\binom{n}{3} + \binom{n}{2} = \binom{n+1}{3}$. □

3 Rectifiers

We introduce the notion of rectifier as an instrument for modifying non-metric dissimilarities into metrics, with the preservation of the relative order of the dissimilarities between objects.

Definition 1. *A* rectifier *is a function* $f : \mathbb{R}_{\geq 0} \times U \longrightarrow \mathbb{R}_{\geq 0}$ *that satisfies the following conditions:*

(i) $U \subseteq \mathbb{R}_{>0}$ *and* $\inf U = 0$;
(ii) $\lim_{\alpha \to 0+} f(t, \alpha) = y_0$ *for every* $t > 0$, *where* $y_0 > 0$;
(iii) $f(0, \alpha) = 0$ *for every* $\alpha \in U$;
(iv) f *is strictly increasing in its first argument;*
(v) f *is sub-additive in its first argument, that is* $f(t_1 + t_2, \alpha) \leq f(t_1, \alpha) + f(t_2, \alpha)$ *for* $t_1, t_2 \in \mathbb{R}_{\geq 0}$ *and* $\alpha \in U$.

The fourth condition of the previous definition is needed to preserve the relative order of the dissimilarities.

Example 1. Let $f(t, \alpha) = t^\alpha$ for $t \in \mathbb{R}_{\geq 0}$ and $\alpha \in (0, 1]$. The function f is a rectifier. Indeed, we have $\lim_{\alpha \to 0+} t^\alpha = 1$ for every $t > 0$ and $f(0, \alpha) = 0$ for every $\alpha \in (0, 1]$.

For a fixed α the function f is obviously strictly increasing in its first argument. Furthermore, for any $t_1, t_2 > 0$ the function $\varphi(\alpha) = \left(\frac{t_1}{t_1+t_2}\right)^\alpha + \left(\frac{t_2}{t_1+t_2}\right)^\alpha$ is decreasing on $[0, 1]$ and $\varphi(1) = 1$. Therefore, $\left(\frac{t_1}{t_1+t_2}\right)^\alpha + \left(\frac{t_2}{t_1+t_2}\right)^\alpha \geq 1$, which yields the sub-additivity.

Example 2. Let $g(t, \alpha) = 1 - e^{-\frac{t}{\alpha}}$ for $t \in \mathbb{R}_{\geq 0}$ and $\alpha \in (0, \infty)$. We claim that g is a rectifier.

Indeed, we have $\lim_{\alpha \to 0+} g(t, \alpha) = 1$ for every $t > 0$. Also, $g(0, \alpha) = 0$ and g is obviously strictly increasing in t. The sub-additivity of g in its first argument amounts to

$$1 - e^{-\frac{(t_1+t_2)}{\alpha}} \leq 2 - e^{-\frac{t_1}{\alpha}} - e^{-\frac{t_2}{\alpha}}, \tag{1}$$

or equivalently $1 - u - v + uv \geq 0$, where $u = e^{-\frac{t_1}{\alpha}}$ and $v = e^{-\frac{t_2}{\alpha}}$. In turn, this is equivalent to $(1 - u)(1 - v) \geq 0$. Since $t \geq 0$, $u, v \leq 1$ which proves the sub-additivity of g.

Note that for a rectifier $f(t, \alpha)$ and a metric d, the function $d_\alpha : S \times S \longrightarrow \mathbb{R}_{\geq 0}$ defined by $d_\alpha(x, y) = f(d(x, y), \alpha)$ is also a metric on S. Indeed, $d(x, y) \leq d(x, z) + d(z, y)$ implies $d_\alpha(x, y) = f(d(x, y), \alpha) \leq f(d(x, z) + d(z, y), \alpha)$ because f is increasing and $f(d(x, z) + d(z, y), \alpha) \leq f(d(x, z), \alpha) + f(d(z, y), \alpha)$ because f is sub-additive. Together, they yield the triangular inequality. However, our interest in the notion of rectifier stems mainly from the following result.

Theorem 2. *Let* $d : S \times S \longrightarrow \mathbb{R}_{\geq 0}$ *be a (in)definite dissimilarity and let* f *be a rectifier. There exists* $\delta > 0$ *such that the function* d_α *is a (semi-)metric on* S *if* $\alpha < \delta$. *Furthermore, if* $d(s_1, s_1') \leq d(s_2, s_2')$, *then* $d_\alpha(s_1, s_1') \leq d_\alpha(s_2, s_2')$ *for* $s_1, s_1', s_2, s_2' \in S$.

Proof: Note that $d_\alpha(x,x) = f(d(x,x),\alpha) = f(0,\alpha) = 0$ due to the third property of f. Also, it is immediate that $d_\alpha(x,y) = d_\alpha(y,x)$, so we need to show only that there exists δ with the desired properties.

Since $\lim_{\alpha\to 0+} f(t,\alpha) = y_0$ for any $t > 0$, it follows that for every $\epsilon > 0$ there is $\delta(\epsilon,t) > 0$ such that $\alpha < \delta(\epsilon,t)$ implies $y_0 - \epsilon < f(t,\alpha) < y_0 + \epsilon$ *for every* $t > 0$. If we choose $\delta_0(\epsilon) = \min\{\delta(\epsilon,t) \mid t \in R(d)\}$, then $\alpha < \delta_0(\epsilon)$ implies $d_\alpha(x,y) = f(d(x,y),\alpha) < y_0 + \epsilon$ and $d_\alpha(x,z) + d_\alpha(z,y) = f(d(x,z),\alpha) + f(d(z,y),\alpha) > 2y_0 - 2\epsilon$. If ϵ is sufficiently small we have $y_0 + \epsilon < 2y_0 - 2\epsilon$, which implies $d_\alpha(x,y) \leq d_\alpha(x,z) + d_\alpha(z,y)$, which concludes the argument for the first part of the statement. The second part follows immediately. □

Theorem 2 shows that by using a rectifier we can transform a dissimilarity into a semi-metric. In some instances we can avoid computing $\delta_0(\epsilon)$ using the technique shown in the next example.

Example 3. Let $f(t,\alpha) = t^\alpha$ be the rectifier considered in Example 1. Suppose that $T_{(w,\{u,v\})} \in \mathsf{NMT}(S)$ that is, $d(u,v) > d(u,w) + d(w,v)$.

Since $d(u,v)^0 \leq d(u,w)^0 + d(w,v)^0$, the set

$$E_{u,v,w} = \{\alpha \in \mathbb{R}_{\geq 0} \mid d(u,v)^\alpha \leq d(u,w)^\alpha + d(w,v)^\alpha\}$$

is non-empty, so $\sup E_{u,v,w} \geq 0$. If $\alpha_S = \inf\{\sup E_{u,v,w} \mid u,v,w \in S\} > 0$, then d_{α_S} is a non-trivial semi-metric on S.

Thus, we need to solve the inequality $1 \leq a^\alpha + b^\alpha$, where

$$a = \frac{d(u,w)}{d(u,v)} \text{ and } b = \frac{d(w,v)}{d(u,v)}.$$

Because of the assumption made about (u,v,w) we have $a + b < 1$, so we have $a, b < 1$.

The solution of this inequality cannot be expressed using elementary functions. However, a lower bound of the set of solution can be obtained as follows.

Let $f : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$ be the function defined by $f(x) = a^x + b^x$. It is clear that $f(0) = 2$ and that f is a decreasing function because both a and b belong to $[0,1)$. The tangent to the graph of f in $(0,2)$ is located under the curve and its equation is

$$y - 2 = x \ln ab.$$

Therefore, an upper bound of the solution of the equation $1 = a^x + b^x$ is obtained by intersecting the tangent with $y = 1$, which yields

$$x = -\frac{1}{\ln ab} = \left(\ln \frac{d^2(u,v)}{d(u,w)d(w,v)}\right)^{-1}.$$

Thus, if

$$\alpha \leq \inf\left\{\left(\ln \frac{d^2(u,v)}{d(u,w)d(w,v)}\right)^{-1} \middle| T_{(w,\{u,v\})} \in \mathsf{NMT}(\mathcal{S})\right\},$$

we can transform d into semi-metric d_α.

Example 4. It is easy to see that for the dissimilarity space (S_{pq}, d_{pq}) introduced in Section 2, the function $f(t, \alpha) = t^{\alpha}$ is a rectifier. If $\alpha \leq \frac{1}{2 \ln \frac{b}{c}}$ this function can be used for transforming (S_{pq}, d_{pq}) to a metric space.

4 Impact of TI Violations on Clustering

We evaluate the impact of using a triangular inequality violating dissimilarity d on clustering. Let $\mathcal{S} = (X, d)$ be a dissimilarity space, where $X = \{x_1, \ldots, x_n\}$. Without loss of generality, we may assume that the range of a dissimilarity has the form $R(d) \subseteq \{n \in \mathbb{N} \mid 0 \leq n \leq m\}$, where $m \in \mathbb{N}$ is the maximum value for dissimilarity d. This is a safe assumption, since we can multiply all the dissimilarities among a finite set of objects by a positive constant without affecting their ratios. Define,

$$\mathbf{AVG}(d) = \sum_{1 \leq i < j \leq n} \frac{2d(x_i, x_j)}{n^2 - n}.$$

Then, if $d(x_i, x_j) \leq \mathbf{AVG}(d)$ we say that x_i, x_j are *almost-similar*, otherwise they are *almost-dissimilar*.

In a non-metric triangle $T_{(x_j, \{x_i, x_k\})}$ the objects x_i, x_k may be similar to x_j but very dissimilar to each other. Clearly, this fact impacts negatively the quality of the clustering. Yet, the degree of impact differs depending on which of the following cases may occur:

1. If x_i, x_k are almost-similar, that is, $d(x_i, x_k) \leq \mathbf{AVG}(d)$, then $d(x_i, x_j) \leq \mathbf{AVG}(d)$ and $d(x_j, x_k) \leq \mathbf{AVG}(d)$. Thus, all three objects are almost-similar to each other and the clustering algorithm will likely place all three objects in one cluster which limits the negative impact of this instance of TI violation.
2. If $d(x_i, x_j) > \mathbf{AVG}(d)$ and $d(x_j, x_k) > \mathbf{AVG}(d)$. then $d(x_i, x_k) > \mathbf{AVG}(d)$. No pair of objects are almost-similar and the clustering algorithm will likely place each object in a separate cluster, which cancels the effects of this triangular inequality violation.
3. If $d(x_i, x_k) > \mathbf{AVG}(d)$, $d(x_i, x_j) > \mathbf{AVG}(d)$ and $d(x_j, x_k) \leq \mathbf{AVG}(d)$, then x_i is almost-dissimilar from the two other objects. The clustering algorithm will likely put the two similar objects x_j and x_k in one cluster and x_i in another. This diminishes the negative influence of this triangular inequality violation.
4. The last case occurs when $d(x_i, x_k) > \mathbf{AVG}(d)$, $d(x_i, x_j) \leq \mathbf{AVG}(d)$ and $d(x_j, x_k) \leq \mathbf{AVG}(d)$. In this situation, if the clustering algorithm assigns all three objects to one cluster, we end up with two almost-dissimilar objects x_i and x_k inside a cluster which is not desirable. On the other hand, if the clustering algorithm puts the two dissimilar objects x_i and x_k in two different clusters and x_j in one of the two clusters, for instance in the cluster which contains x_k then, two almost-similar objects x_i and x_j are in two different clusters which is also undesirable. Thus, in this case the impact of triangular violation is substantial.

We penalize the dissimilarity for any triangular inequality violation, but this penalty must be heavier on instances of the last case. Define the number

$$\theta_{ijk} = \mathbf{AVG}(d) \cdot \max\left(\frac{d(x_i,x_k) - \mathbf{AVG}(d)}{d(x_i,x_j) + d(x_j,x_k)}, 0\right).$$

Let $T_{(x_j,\{x_i,x_k\})}$ be a non-metric triangle. If the TI violation falls in to the first category, then $\theta_{ijk} = 0$. If the violation falls into second and third categories θ_{ijk} will be a positive number. For the last case, θ_{ijk} will be a larger positive number which exhibits the negative impact of this violation on clustering. To make the magnitude of θ_{ijk} consistent across different values of m we introduce its normalized version

$$\begin{aligned}\hat{\theta}_{ijk} &= \frac{2\theta_{ijk}}{\mathbf{AVG}(d)(m - \mathbf{AVG}(d))} \\ &= \max\left(\frac{2\ d(x_i,x_j) - 2\ \mathbf{AVG}(d)}{(d(x_i,x_q) + d(x_j,x_q))(m - \mathbf{AVG}(d))}, 0\right).\end{aligned}$$

The total score for the impact of TI violations of d on clustering is defined as $\Phi(X,d) = \sum\{\hat{\theta}_{ijk} \mid T_{(x_j,\{x_i,x_k\})} \in \mathsf{NMT}(\mathcal{S})\}$.

$\Phi(X,d)$ is normalized by dividing it by the maximum possible number of triangular inequality violations in order to make the magnitude of measure consistent across different values of n, the number of objects: $\hat{\Phi}(X,d) = \frac{6\Phi(X,d)}{n(n-1)(n-2)}$.

5 A Quality Measure for Clusterings

Let $\mathcal{C} = \{C_1, C_2, \cdots, C_k\}$ be a clustering of the set of objects S and assume that m_i is the medoid of the cluster C_i for $1 \leq i \leq k$. To assess the quality of the clustering $\mathcal{C}$ we define a measure which we refer as *coherence degree* of the clustering and is denoted by $\gamma(\mathcal{C})$. This measure is computed from a $k \times k$ matrix $\mathcal{I}$ referred to as the *incoherence matrix* defined as follows:

$$\mathcal{I}_{ij} = \begin{cases} \frac{\max_{v \in C_i} \sum_{u \in C_i} d(u,v)}{\sum_{u \in C_i} d(u,m_i)} & \text{if } i = j \text{ and } |C_i| > 1, \\ 1 & \text{if } i = j \text{ and } |C_i| = 1, \\ \frac{\sum_{v \in C_j} d(m_i,v)}{\min_{u \in C_i} \sum_{v \in C_j} d(u,v)} & \text{otherwise.} \end{cases}$$

In a "good" clustering, objects within a cluster are similar to each other, and objects that belong to distinct clusters are dissimilar. We construct clusters based on the similarity of objects to medoids. Thus, a clustering is considered as coherent if the average dissimilarity between an object and the other members of the cluster is about the same as the average dissimilarity between the medoid of the cluster and the non-medoid objects of the cluster. The diagonal elements of $\mathcal{I}$ measure the lack of this coherence property of clusters. Also, the sum of dissimilarities between an object $u \in C_i$ and all objects $v \in C_j$ should be about the same as the sum of dissimilarities of m_i from all objects $v \in C_j$. The off-diagonal elements of $\mathcal{I}$ measure the absence of this coherence quality.

This allows us to define *the coherence degree of a clustering* $\mathcal{C}$, $\gamma(\mathcal{C})$, as

$$\gamma(\mathcal{C}) = \left(\frac{\mathsf{trace}(\mathcal{I})}{2k} + \frac{\sum_{i \neq j} \mathcal{I}_{ij}}{2(k^2 - k)} \right)^{-1}.$$

The measure $\gamma(\mathcal{C})$ is biased in favor of clusterings with large number of clusters because when k is large, the clustering algorithm has more freedom to cover up the impact of TI violations.

6 Experimental Results

We performed two series of experiments. In the first series we randomly generated symmetric $n \times n$ dissimilarity matrices with the maximum dissimilarity value m. For such a matrix $\mathcal{M}$ the corresponding dissimilarity is denoted by $d_{\mathcal{M}}$. In the next step, we applied the PAM clustering algorithm to partition the set of n objects into k clusters using $d_{\mathcal{M}}$. We computed the coherence degree $\gamma(\mathcal{C}_{\mathcal{M}})$ for the resulting clustering $\mathcal{C}_{\mathcal{M}}$ and $\hat{\Phi}(X, d_{\mathcal{M}})$ for dissimilarity $d_{\mathcal{M}}$.

This process was repeated 200 times for randomly generated dissimilarity matrices such that the number $\hat{\Phi}(X, d_{\mathcal{M}})$ lies within a given subinterval. Figure 1 shows the results of this experiment. The x-coordinate of each point is the $\hat{\Phi}(X, d_{\mathcal{M}})$ average over dissimilarities and the y-coordinate is the average coherence degrees $\gamma(\mathcal{C}_{\mathcal{M}})$ for the clusterings of the form $\mathcal{C}_{\mathcal{M}}$. A clear descending trend in the coherence degree of resultant clusterings (indicating a deterioration of the quality of these clusterings) occurs as the TI violations measure $\hat{\Phi}(X, d_{\mathcal{M}})$ for underlying dissimilarities $d_{\mathcal{M}}$ increases. In the second experiment, we used the family of dissimilarities d_{pq} described in Section 2, where p specifies the number of non-metric points and $q = n - p$ the number of metric points. The parameters a, b and c were set to $1, 7$ and 3, respectively, and we varied the values of p. Then, we rectified the dissimilarities d_{pq} by applying the rectifier described in Example 4 resulting in the rectified dissimilarities d^r_{pq}. Then, we used

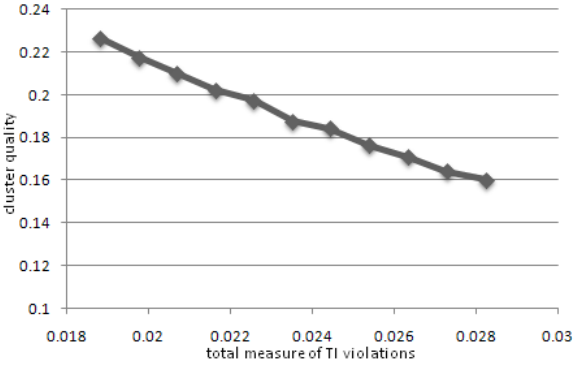


Fig. 1. $k = 7$, $n = 60$ and $m = 100$

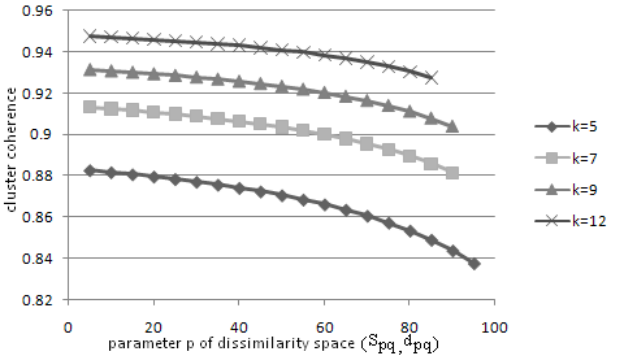


Fig. 2. Plot of $\gamma(\mathcal{C}_{pq})$ to p for $n = 100$

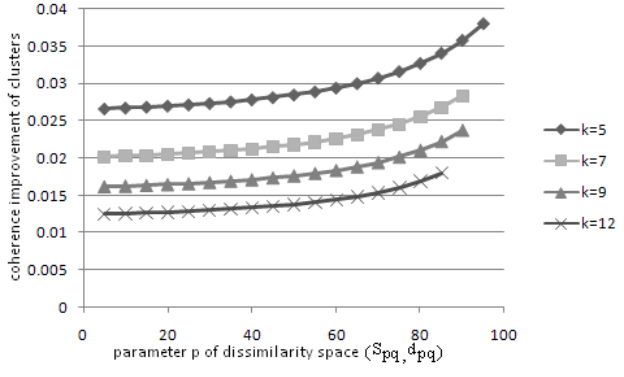


Fig. 3. Plot of $\gamma(\mathcal{C}^r_{pq}) - \gamma(\mathcal{C}_{pq})$ to p for $n = 100$

PAM to generate the clusterings $\mathcal{C}_{pq}$ and $\mathcal{C}^r_{pq}$ based on these two dissimilarities, respectively and we calculated the coherence measures $\gamma(\mathcal{C}_{pq})$ and $\gamma(\mathcal{C}^r_{pq})$.

Figure 2 shows the decrease in the coherence measure $\gamma(\mathcal{C}_{pq})$ as we increase p for various k and n. That is, the coherence degree of the clustering is decreasing as the number of triangular inequality violations of our dissimilarity increases. Note that we chose numbers for parameters a, b and c such that the fractions $\frac{b}{a}$ and $\frac{b}{c}$ are small. The decrease in coherence of clusters would have more accentuated with an increase in p, if we would have chosen parameters such that the above fractions were larger. Figure 3 shows the difference $\gamma(\mathcal{C}^r_{pq}) - \gamma(\mathcal{C}_{pq})$ as p varies. Observe that not only using rectified dissimilarity yields a clustering with better quality according to coherence measure, but also this improvement in the coherence of the clustering due to rectification process increases as the number of non-metric points increases.

7 Conclusions and Future Work

We investigated the impact of using non-metric dissimilarities in medoid-based clustering algorithms on the quality of clusterings and demonstrated the impact that TI violations have on clustering quality.

In principle, several rectifiers may exists for a dissimilarity space. By quantifying the goodness of a rectifier, a comparative study of classes of rectifiers should be developed.

A similar study will be carried out on several variations of the k-means algorithm, which is centroid-based, as well as on density-based clusterings such as DBSCAN. In the later type of algorithms the notion of density is closely tied with the idea of ϵ-neighborhood of an object. Clearly, objects x and z are in the $\max[d(x,y), d(y,z)]$-neighborhood of y and we expect x and z be in $(d(x,y) + d(y,z))$-neighborhood of each other, which may not be the case in a TI violating triangle $T_{(y,\{x,z\})}$.

References

1. Elkan, C.: Using the triangle inequality to accelerate k-means. In: Proceedings of the 20th International Conference on Machine Learning, pp. 147–153 (2003)
2. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data – An Introduction to Cluster Analysis. Wiley Interscience, New York (1990)
3. Kryszkiewicz, M., Lasek, P.: TI-DBSCAN: Clustering with DBSCAN by means of the triangle inequality. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 60–69. Springer, Heidelberg (2010)
4. Liao, Y., Kaafar, M., Gueye, B., Cantin, F., Geurts, P., Leduc, G.: Detecting triangle inequality violations in internet coordinate systems. In: Fratta, L., Schulzrinne, H., Takahashi, Y., Spaniol, O. (eds.) NETWORKING 2009. LNCS, vol. 5550, pp. 352–363. Springer, Heidelberg (2009)
5. Lumezanu, C., Baden, R., Spring, N., Bhattacharjee, B.: Triangle inequality and routing policy violations in the internet. In: Moon, S.B., Teixeira, R., Uhlig, S. (eds.) PAM 2009. LNCS, vol. 5448, pp. 45–54. Springer, Heidelberg (2009)
6. Moore, A.W.: The anchors hierarchy: Using the triangle inequality to survive high dimensional data. In: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, pp. 397–405. AAAI Press, Menlo Park (2000)
7. Pan, J.S., McInnes, F.R., Jack, M.A.: Fast clustering algorithms for vector quantization. Pattern Recognition 29, 511–518 (1966)

Batch Weighted Ensemble for Mining Data Streams with Concept Drift

Magdalena Deckert

Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, Poland
magdalena.deckert@cs.put.poznan.pl

Abstract. This paper presents a new framework for dealing with two main types of concept drift: sudden and gradual drift in labelled data with decision attribute. The learning examples are processed in batches of the same size. This new framework, called Batch Weighted Ensemble, is based on incorporating drift detector into the evolving ensemble. Its performance was evaluated experimentaly on data sets with different types of concept drift and compared with the performance of a standard Accuracy Weighted Ensemble classifier. The results show that BWE improves evaluation measures like processing time, memory used and obtain competitive total accuracy.

1 Introduction

Most of the existing classifiers extract knowledge from static data. Those classifiers fail to answer modern challenges in classification like processing streaming data. Data streams are characterized by large size of data, probably infinite [6]. I assume that the learning data are labelled with true decision class value and they arrive in batches of the same size. In real life situations true decision label may not be known immediately, but may be delayed in time. There exists other approaches that process labelled and unlabelled data streams like [13]. One of the problem with processing data streams is that the environment and classification problem may change in time. The concepts of interest may depend on some hidden context [11]. One of the common examples is detecting and filtering out spam e-mail. The description of assignment to different groups of e-mails changes in time. They depend on user preferences and active spammers, who are inventing new solutions to trick the up-to-date classifier. Changes in the hidden context can induce more or less radical changes in target concepts, producing what is known as *concept drift* [9].

The term of concept drift was introduced by Schlimmer and Granger in [8]. It means that the statistical properties of the decision class change over time in unforeseen ways. This causes problems because the predictions become less accurate with time. There exists two main types of change: *sudden (abrupt)* and *gradual (incremental)* [9]. For example John was listening to pop music his whole teenage life but when he graduated from the university he changed his preferences and started to listen only to classical music. This is example of

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 290–299, 2011.

sudden drift. Gradual drift would occur if John would start to listen to classical music while he was still enjoying pop music but the interest in pop decreases with time.

Mining data streams in presence of concept drift is a rather new discipline in machine learning world but there exist some algorithms that solve this problem. They can be divided into two main groups: trigger based and evolving. Trigger based model contains a change detector, which indicates a need for model change. The change-detection process is separate from the classification. Standard actions of classifiers, equiped with a detector, are as following: the classifier predicts a label for recived instance x; then the true label and the predicted label are submitted to the change detector; if change is detected the classifier is re-trained [7]. Evolving methods operate in a different way. They try to build the most accurate classifiers at each moment of time without explicit information about occurrence of change. The most popular evolving technique for handling concept drift is an *ensemble of classifiers* [12]. They are naturally adjusted to processing data in blocks.

According to Wang et al. a simple ensemble might be easier to use in changing environments than a single adaptive classifier. They proposed an ensemble of classifiers called Accuracy Weighted Ensemble (AWE). Their study showed that AWE combines advantages of a set of experts and adjusts quite well to the underlying concept drift [10]. One of the main disadvantage of evolving ensemble of classifiers is that it builds a new base classifier for every batch of new data. This results in high usage of memory. Tests conducted by Brzezinski [3] showed that AWE has problems with well adjusting to gradual drift and for strong sudden drift it deletes all of the base classifiers and starts building ensemble from scratch.

In my opinion those are undesirable features and they should be improved. In order to do that I propose to introduce a change detector into an evolving ensemble of classifier. Change detectors are naturally suitable for the data where sudden drift is expected. Ensembles, on the other hand, are more flexible in terms of change type, while they can be slower in reaction in case of sudden drift [12]. Thanks to combination of the two abovementioned components we can obtain ensemble, which is well adjusted to both main types of changes. Moreover this ensemble would not build a new classifier for each batch of the data when the concepts are stable. Next adavantage of such a cooperation is that direct indication of the type of change can cause better reaction of ensemble.

The main aim of this paper is to present the new framework for dealing with two main types of concept drift: sudden and gradual drift. This new framework is based on incorporating a drift detector into an evolving ensemble. Its performance was evaluated experimentaly on data sets with different types of concept drift and compared with performance of the standard AWE classifier. Evaluation criteria, on which ensembles will be compared, are: total accuracy of classification, use of memory and processing time.

This paper is organized as following. The next section presents related works in detecting concept drift and creating evolving ensembles. In section 3 a new

framework for building Batch Weighted Ensemble is presented. In section 4 I present experimental evaluation of the proposed framework. Section 5 concludes this paper.

2 Related Works

In this section I will concentrate on methods that are most related to my study (for reviews of other approaches see [5,6,7,9,12]). As it was said in the introduction, methods that deal with problem of concept drift can be divided into two main groups: trigger based and evolving.

One of the most popular drift detection method is DDM proposed by Gama et al. in [4]. This approach detects changes in the probability distribution of examples. The main idea of this method is to monitor the error-rate produced by a classifier. Statistical theory affirm that error decreases if the distribution is stable [4]. When the error increases, it signifies that the distribution has changed. DDM operates on labelled data that arrive one at a time. For each example, the predicted class value can be True or False, so at each moment of time t, the error-rate is the probability of observing False p_t, with standard deviation s_t resulting from the Binominal distribution. A significant increase in the error suggests a change in the source distribution, which implies that current classifier is out-of-date. DDM manages two records: p_{min} and s_{min}. Those values are updated whenever the actual value of $p_t + s_t$ is lower than $p_{min} + s_{min}$. DDM signals two levels of change: warning and drift. Warning level is used to define the optimal size of current concept window with learning examples. The warning level is reached if $p_t + s_t \geq p_{min} + 2 * s_{min}$. A new current concept window is declared starting in example when warning level was achived t_w. Learning examples are remembered until drift level is reached. The drift level is achived when $p_t + s_t \geq p_{min} + 3 * s_{min}$. A new model is updated with examples starting in t_w till t_d. It is possible to observe warning level, followed by a decrease in error rate. This situation is treated as a *false alarm* and the model is not refined. This method is independent from the learning algorithm and can be easily extended to processing data in batches.

Processing batches of data is the natural environment for evolving ensembles of classifiers. An example of such an ensemble is Accuracy Weighted Ensemble (AWE) proposed by Wang et al. in [10]. It builds a new base classifier for each incoming batch of data and uses that data to evaluate all the existing ensemble members to select the best component classifiers. The best base classifiers are chosen according to weights calculated for each classifier. Classifiers are weighted by their expected classification accuracy on the test data. To properly weight the component classifiers we need to know the actual function being learned, which is unavailable. Therefore the authors propose to compute values of classifiers' weights by estimating the error rate on the most recent data chunk. Each weight w_i is calculated with formula $w_i = MSE_r - MSE_i$, where MSE_r is mean square error of random classifier ans MSE_i is mean square error of classifier i. The MSE_r can be expressed by $MSE_r = \sum_c p(c) * (1 - p(c))^2$,

where $p(c)$ is distribution of class c. The MSE_i can be expressed by $MSE_i = \frac{1}{|S_n|} \sum_{(x,c) \in S_n} (1 - f_c^i(x))^2$, where $f_c^i(x)$ is the probability given by classifier i that x is an instance of class c. AWE is sensitive to the defined size of a batch. The pseudo-code for the Accuracy Weighted Ensemble is shown in Algorithm 1.

Algorithm 1. Accuracy Weighted Ensemble

Input : S: a data stream of examples; bs: size of the data batch; k: the total number of classifiers; C: a set of k previously trained classifiers
Output: C: a set of classifiers with updated weights

foreach *data block* $b_i \in S$ **do**
  train classifier C' on b_i;
  compute error rate of C' via cross validation on b_i;
  derive weight w' for C' using formula $w' = MSE_r - MSE_i$;
  foreach *classifiers* $C_i \in C$ **do**
    apply C_i on b_i to derive MSE_i;
    compute w_i using formula $w_i = MSE_r - MSE_i$;
  $C \leftarrow k$ of the top weighted classifiers in $C \cup C'$;
Return C

3 Batch Weighted Ensemble Framework

In my framework I propose to introduce a change detector into an evolving ensemble of classifier. Change detectors are naturally suitable for the data with sudden drift. Ensembles, on the other hand, easily adapt to gradual change but can be slower in reaction in case of sudden drift. Thanks to combination of the two abovementioned components we can obtain ensemble, which is well adjusted to both main types of changes. Moreover this ensemble would not build new classifiers for batches with stable concepts. This will result in lower usage of memory.

Proposed Batch Weighted Ensemble (shortly called BWE) contains drift detection method operating on batches of data. This detector, called Batch Drift Detection Method (BDDM), is presented as Algorithm 2. At first BDDM is building a table with values of accumulated classification accuracy for each learning example. Then a regression model is build on the previously calculated values. I use a simple linear regression in order to find a trend in the data. This simplification will not find ideal adjustment to the data, it only estimates a tendency in the data. Thanks to the use of regression, BDDM will not process every batch but only those with decreasing trend. Next advantage is that regression more clearly shows direction of change. Statistics without regression may be more sensitive to the outliers. The second step of BDDM is to discover level of change. BDDM detects two levels of change: warning and drift, according to the formulas given in BDDM's pseudo-code.

Batch Drift Detection Method is embedded into Batch Weighted Ensemble. When the ensemble is empty, BWE builds a new base classifier on current batch of data. In the opposite case, BWE launches BDDM to find trend in the current

Algorithm 2. Batch Drift Detection Method

Input : B: a batch of examples; C: an ensemble of classifiers
Output: *signal*: flag indicating type of discovered signal

foreach *example* $b_i \in B$ **do**
  calculate accumulated accuracy with C;
create regression function on accumulated accuracy;
if $a<0$ **then** $\Longleftarrow$ test regression gradient parameter a
  if $(average.accuracy - standard.deviation < maxaccuracy - 2 * standard.deviation)$ **then**
    $signal$ = warning;
  if $(average.accuracy - standard.deviation < maxaccuracy - 3 * standard.deviation)$ **then**
    $signal$ = drift;
Return *signal*

batch of learning examples. If BDDM signals warning level, a new classifier is build on the batch and added to the ensemble. If detector signals drift, again a new classifier is build on the batch and added to the ensemble. Next, weights for each component classifier are established. In the end, BWE removes every base classifier if its weight equals 0. If ensemble deletes all components, a classifier on last batch of the data is added. Expanded version of BWE framework, in case of zero size of the ensemble, builds 10 base classifiers on bootstrap samples created from the last batch of the data. A simple version of BWE is presented as Algorithm 3.

4 Experiments

The main aim of my experiments was to evaluate efficiency of proposed framework on measures like processing time, memory used and accuracy of classification. Thus, I compared the performance of base BWE algorithm and BWE with bootstrapping to the performance of the standard AWE ensemble.

All base classifiers were constructed using C4.5 (J48 from WEKA) – as an algorithm of induction of decision trees. Algorithms were run without prunning in order to get more precise description of current batch of data. Ensemble algorithms were implemented in Java and were embedded into Massive Online Analysis framework. MOA is a framework for data stream mining. It contains a collection of machine learning algorithms, data generators and tools for evaluation. More about this project can be found in literature [1,2] and on MOA project website [1]. MOA can be easily extended with new mining algorithms, but also with new stream generators or evaluation measures. In my experiments I used extentions for processing and evaluation batches of data implemented by Brzezinski. Full description of this extention is available in [3].

[1] see: http://moa.cs.waikato.ac.nz/

Algorithm 3. Batch Weighted Ensemble

Input : S: a data stream of examples; b: size of the data batch; C: a set of previously trained classifiers
Output: C: an updated set of classifiers

```
foreach data chunk b_i ∈ S do
    if size of ensemble = 0 then
        train classifier C' on b_i;
        C ← C';
    else
        BDDM (C, b_i); {build Batch Drift Detection Method on batch b_i of data}
        if signal=warning then
            if memory buffer is full then
                remove classifier with the lowest weight;
            train classifier C' on b_i;
            C ← C ∪ C';
        else if signal=drift then
            if memory buffer is full then
                remove classifier with the lowest weight;
            train classifier C' on b_i;
            C ← C ∪ C';
            foreach classifiers C_i ∈ C do
                compute w_i using a formula: w_i = MSE_r − MSE_i;
                if w_i = 0 then
                    remove classifier C_i;
            if size of ensemble = 0 then
                train classifier C' on b_i;
                C ← C';
Return C
```

All experiments were carried out on 3 different data sets: elec2, hyperplane and STAGGER. Electricity is a real data set containing energy prices from the electricity market in the Australian state of New South Wales. These prices were affected by market demand, supply, season, weather and time of day. From the original data set I selected only a time period with no missing values. Hyperplane is a popular generator that creates streams with gradual concept drift by rotating the decision boundary for each concept. STAGGER is a data generator that creates streams with sudden concept drift. In my data set sudden drift occurs every 10000 examples. Full description of STAGGER problem can be found in [8]. Detailed characteristics of this datasets are given in Table 1. I chose them also because they were often used by other researchers working with concept drift.

Ensembles were run with standard parameters. For AWE number of the best classifiers was set to 15 and the memory buffer was restricted to 20 classifiers. BWE memory buffer was set to the same value for better comparison. I tested three different sizes of batch of data: 500, 1000 and 1500. Unfortunately be-

Table 1. Characteristics of data sets

Data set	Objects	Attributes	Classes
elec2	27552	6	2
hyperplane	30000	10	2
STAGGER	30000	3	2

Table 2. Average total accuracy

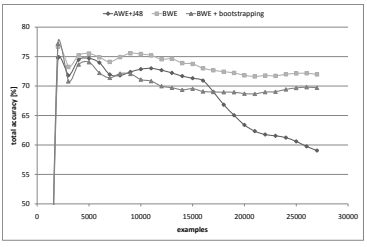
Data set	Classifier		
	AWE	BWE	BWE + bootstrapping
elec2	65.92	70.02	68.75
hyperplane	65.36	70.80	72.08
STAGGER	88.97	91.66	91.66

cause of lack of space I present summary of average total accuracy for different classifiers and data sets in Table 2.

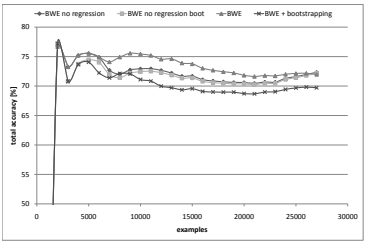
I tested effectiveness of ensembles after every batch of data. Besides total accuracy I recorded also values of accumulated processing time from the begining of learning phase and size of current model. Results of processing time, memory used and total accuracy for elec2 data set are presented on Figures: 1(a), 2(a) and 3(a). Total accuracy for hyperplane and STAGGER data sets are presented in Figures: 1(c) and 1(d). I do not present results of processing time and memory on hyperplane and STAGGER data sets because they are similar to the results obtained on elec2 data. I analyze changes that occure with time, that is why results presented on figures are more comprehensive than in tables. Because of lack of space I present figures for batch size = 1000. Results for other sizes of batch obtained on memory and processing time are analogical. However value of total accuracy varies with size of the batch. For elec2 and STAGGER dataset smaller batch size implies better accuracy. For hyperplane dataset results are

Table 3. Hits statistics for BWE algorithms

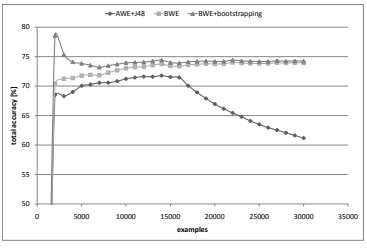
Data set	Hit type	Classifier			
		regression		no regression	
		BWE	BWE + bootstrapping	BWE	BWE + bootstrapping
elec2	warning	1	1	2	3
	drift	15	9	24	23
	processed	16	10	26	26
hyperplane	warning	0	0	1	1
	drift	21	21	26	26
	processed	21	21	29	29
STAGGER	warning	0	0	0	0
	drift	2	2	2	2
	processed	2	2	29	29



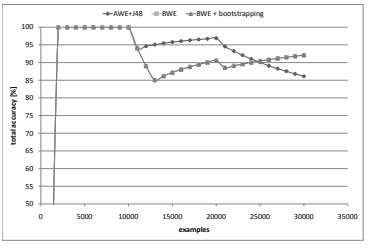
(a) elec2



(b) elec2-regression

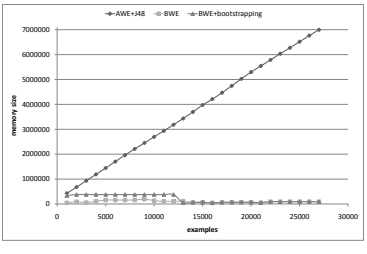


(c) hyperplane

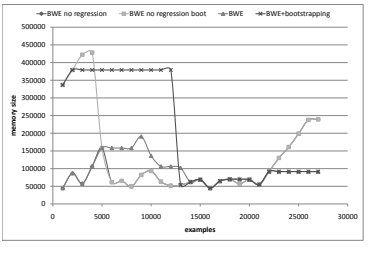


(d) STAGGER

Fig. 1. Results for accuracy of classification



(a) elec2

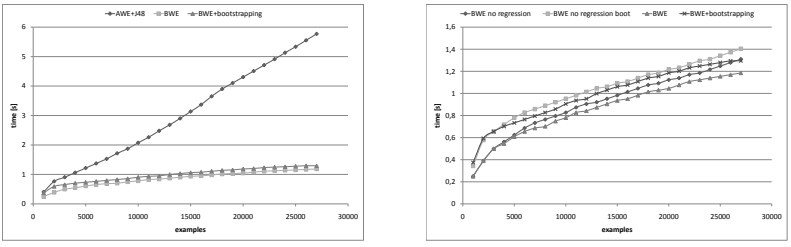


(b) elec2-regression

Fig. 2. Results for memory usage

slightly better for larger size of the batch. This behaviour derives from the type of the drift. Smaller batch size is more appropriate for sudden concept drift, because it allows faster reaction to the change. On the other hand, larger batch size is more adequate for gradual drift. Thanks to more examples in the batch, regression model can better adjust to incremental change, which will reflect in higher accuracy.

For better insight into BWE framework I present statistics from Batch Drift Detection Method. In Table 3 are presented statistics on numbers of detected warning or drift signals and numbers of processed batches for batch size = 1000.



(a) elec2 (b) elec2-regression

Fig. 3. Results for processing time

Use of regression decreases number of batches processed by BDDM. Number of processed batches depends on the type of change, that occurs in the dataset. The number for sudden drift is significantly smaller than for gradual drift. I also tested influence of the linear regression on interesting measures. Results showed that the regression helps to decrease size of used memory and processing time, while accuracy of classification is kept on similar level. Sample results of regression usage comparison are presented on figures: 1(b), 2(b) and 3(b).

5 Discussion of Results and Final Remarks

In this paper I presented a new framework for dealing with two main types of concept drift: sudden and gradual drift in labelled data. The learning examples are processed in batches of the same size with a framework called Batch Weighted Ensemble. It is based on incorporating drift detector into the evolving ensemble.

After comparing exact results of BWE to AWE on memory usage one can notice that AWE uses always more memory. The reason of this behaviour is that AWE builds a new base classifier on every batch of the data. Of course the number of components is limited to buffer size, but results showed that size of the model is still growing. BWE thanks to embedded drift detector does not build a base classifier for every batch. This can be seen in the number of detected warning and drift signals. The drift detector does not fire for every chunk of the data, so a new component is not created every time. For STAGGER data set BWE found two drifts. This is correct behaviour, because concept has changed suddenly two times. After every 10000 examples the total accuracy starts decreasing and then the method discovers drift. Comparison of the two versions of BWE showed that they build models of similar size.

Results achived on processing time are similar to those obtained on memory usage. Both BWE ensembles are much faster than AWE classifier.

Comparison of total accuracy showed that, for gradual drift in hyperplane data set, BWE can achive better results than AWE framework. On STAGGER data set, in which sudden drift occurs every 10000 examples, AWE learns faster for first change but in the end BWE achives higher value of total accuracy.

To sum up, the experimental evaluation on 3 data sets with different types of drift showed that BWE improves evaluation measures like processing time, memory used and obtain competitive total accuracy.

My future research in processing data with concept drift covers improvement in the drift detector. It reacts slower for the first change in STAGGER data set. This can be enhanced. Next topic of my work is an algorithm that can also deal with reoccuring concepts. It should also recognize some unwanted events like noise or blips. This is the subject of my on-going research.

References

1. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive Online Analysis. Journal of Machine Learning Research, JMLR (2010), http://sourceforge.net/projects/moa-datastream/
2. Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., Seidl, T.: MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. In: Workshop on Applications of Pattern Analysis, HaCDAIS (2010)
3. Brzezinski, D.: Mining data streams with concept drift. Master's thesis supervised by J.Stefanowski, Poznan University of Technology, Poznań, Poland (2010)
4. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with Drift Detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
5. Gama, J.: Knowledge Discovery from Data Streams. CRC (2010)
6. Kuncheva, L.I.: Classifier Ensembles for Changing Environments. In: Roli, F., Kittler, J., Windeatt, T. (eds.) MCS 2004. LNCS, vol. 3077, pp. 1–15. Springer, Heidelberg (2004)
7. Kuncheva, L.I.: Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. In: Proceedings 2nd Workshop SUEMA 2008 (ECAI 2008), Greece, pp. 5–10 (2008)
8. Schlimmer, J., Granger, R.: Beyond Incremental Processing: Tracking Concept Drift. AAAI, Menlo Park (1986)
9. Tsymbal, A.: The problem of concept drift: Definitions and related work. Technical Report, Trinity College, Dublin, Ireland (2004)
10. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proceedings ACM SIGKDD, pp. 226–235 (2003)
11. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23, 69–101 (1996)
12. Zliobaite, I.: Learning under Concept Drift: an Overview. Technical Report, Vilnius University, Lithuania (2009)
13. Zhang, P., Zhu, X., Guo, L.: Mining Data Streams with Labeled and Unlabeled Training Examples. In: Perner, P. (ed.) ICDM 2009. LNCS, vol. 5633, pp. 627–636. Springer, Heidelberg (2009)

A Generic Approach for Modeling and Mining n-ary Patterns

Mehdi Khiari, Patrice Boizumault, and Bruno Crémilleux

GREYC, CNRS - UMR 6072, Université de Caen Basse-Normandie,
Campus Côte de Nacre, F-14032 Caen Cedex, France
{Forename.Surname}@info.unicaen.fr

Abstract. The aim of this paper is to model and mine patterns combining several local patterns (n-ary patterns). First, the user expresses his/her query under constraints involving n-ary patterns. Second, a constraint solver generates the correct and complete set of solutions. This approach enables to model in a flexible way sets of constraints combining several local patterns and it leads to discover patterns of higher level. Experiments show the feasibility and the interest of our approach.

1 Introduction

Knowledge Discovery in Databases involves different challenges, such as the discovery of patterns of a potential user's interest. The constraint paradigm brings useful techniques to express such an interest. If mining local patterns under constraints is now a rather well-mastered domain including generic approaches [1], these methods do not take into account the interest of a pattern with respect to the other patterns which are mined. In practice, a lot of patterns which are expected by the data analyst (cf. Section 2.2) require to consider simultaneously and to combine several patterns. In the following, such patterns are called *n-ary patterns*, and a query involving n-ary patterns is called a *n-ary query*.

There are very few attempts on mining n-ary patterns and the existing methods tackle particular cases by using devoted techniques [8,9]. One explanation of the lack of generic methods is likely the difficulty of the task. Mining n-ary patterns requires to compare the solutions satisfying each pattern involved in the constraint, it is drastically harder than mining local patterns. The lack of generic methods restrains the discovery of useful patterns because the user has to develop a new method each time he wants to extract a new kind of patterns.

In this paper, we propose a generic approach for modeling and mining n-ary patterns using Constraint Programming (CP). Our approach proceeds in two steps. First, the user specifies the set of constraints which has to be satisfied. Such constraints handle set operations and also numeric properties such as the frequency or the length of patterns. Then, a constraint solver generates the correct and complete set of solutions. The great advantage of this modeling is its flexibility, it enables us to define a large set of n-ary queries leading to discover patterns of higher level. It is no longer necessary to develop algorithms from scratch to mine new types of patterns.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 300–305, 2011.

2 Definitions and First Examples

2.1 Local Patterns

Let $\mathcal{I}$ be a set of distinct literals called items, an itemset (or *pattern*) is a non-null subset of $\mathcal{I}$. The language of itemsets corresponds to $\mathcal{L}_{\mathcal{I}} = 2^{\mathcal{I}} \backslash \emptyset$. A *transactional dataset* $\boldsymbol{r}$ is a multi-set of itemsets of $\mathcal{L}_{\mathcal{I}}$. Each itemset, usually called a *transaction* or object, is a database entry. Constraint-based mining task selects all the itemsets of $\mathcal{L}_{\mathcal{I}}$ present in $\boldsymbol{r}$ and satisfying a predicate which is named *constraint*. *Local patterns* are regularities that hold for a particular part of the data (i.e., checking whether a pattern satisfies or not a constraint can be performed independently of the other patterns holding in the data).

Example. Let X be a local pattern. The well-known *frequency* constraint focuses on patterns occurring in the database a number of times exceeding a given minimal threshold: $freq(X) \geq minfr$. There are many other constraints [7] to evaluate the relevance of patterns, like the *area* ($area(X)$ is the product of its frequency times its length: $area(X) = freq(X) \times length(X)$).

2.2 N-ary Patterns

In practice, the data analyst is often interested in discovering richer patterns than local patterns. The definitions relevant to such more complex patterns rely on properties involving several local patterns and are formalized by the notions of *n-ary constraint* and *n-ary pattern* leading to *n-ary queries*.

Definition 1 (n-ary pattern). A *n-ary pattern* is defined by a query involving several patterns.

Definition 2 (n-ary query). A *n-ary query* is a set of constraints over n-ary patterns.

2.3 Motivating Example

N-ary queries straightforwardly enable us to design rich patterns requested by the users such as the discovery of pairs of exception rules without domain-specific information [9]. An exception rule is defined as a pattern combining a strong rule and a deviational pattern to the strong rule, the interest of a rule of the pattern is highlighted by the comparison with the other rule. The comparison between rules means that these exception rules are *not* local patterns. More formally, an exception rule is defined within the context of a pair of rules as follows (I is an item, for instance a class value, X and Y are local patterns):

$$e(X \rightarrow \neg I) \equiv \begin{cases} true & \text{if } \exists Y \in \mathcal{L}_{\mathcal{I}} \text{ such that } Y \subset X, \text{one have } (X \backslash Y \rightarrow I) \wedge (X \rightarrow \neg I) \\ false & \text{otherwise} \end{cases}$$

Such a pair of rules is composed of a common sense rule $X \backslash Y \rightarrow I$ and an exception rule $X \rightarrow \neg I$ since usually if $X \backslash Y$ then I. The exception rule isolates surprising information. This definition assumes that the common sense rule has a high frequency and a rather high confidence and the exception rule has a

low frequency and a very high confidence (the confidence of a rule $X \rightarrow Y$ is $freq(X \cup Y)/freq(X)$). Suzuki proposes a method based on sound pruning and probabilistic estimation [9] to extract the exception rules, but this method is devoted to this kind of patterns.

2.4 Related Work

There are a lot of works to discover local patterns under constraints [7] but there are not so many methods to combine local patterns: pattern teams [6], constraint-based pattern set mining [3] to name a few. Even if these approaches explicitly compare patterns between them, they are mainly based on the reduction of the redundancy or specific aims such as classification processes. Our work is in the new trend on investigations of relationships between data mining and constraint programming [2,4].

3 Modeling and Mining n-ary Queries Using CP

3.1 Examples of n-ary Queries

Exception Rules. (see Section 2.3). Let X and Y be two patterns. Let I and $\neg I \in \mathcal{I}$. Let $minfr, maxfr, \delta_1, \delta_2 \in \mathbb{N}$. The exception rule n-ary query is formulated as it follows:

- $X \backslash Y \rightarrow I$ is expressed by the conjunction: $freq((X \setminus Y) \sqcup [1] I) \geq minfr \wedge (freq(X \setminus Y) - freq((X \setminus Y) \sqcup I)) \leq \delta_1$ ($X \backslash Y \rightarrow I$ is a frequent rule having a high confidence value).
- $X \rightarrow \neg I$ is expressed by the conjunction: $freq(X \sqcup \neg I) \leq maxfr \wedge (freq(X) - freq(X \sqcup \neg I)) \leq \delta_2$ ($X \rightarrow \neg I$ is a rare rule having a high confidence value).

$$exception(X, Y, I) \equiv \begin{cases} freq((X \setminus Y) \sqcup I) \geq minfr \wedge \\ freq(X \setminus Y) - freq((X \setminus Y) \sqcup I) \leq \delta_1 \wedge \\ freq(X \sqcup \neg I) \leq maxfr \wedge \\ freq(X) - freq(X \sqcup \neg I) \leq \delta_2 \end{cases}$$

Unexpected Rules. Another example of n-ary queries is the *unexpected* rule $X \rightarrow Y$ with respect to a belief $U \rightarrow V$ where U and V are patterns [8]. Basically, an unexpected rule means that Y and V logically contradict each other. It is defined more formally as: (1) $Y \wedge V \vDash False$, (2) $X \wedge U$ holds (it means XU frequent), (3) $XU \rightarrow Y$ holds ($XU \rightarrow Y$ frequent and has a sufficient confidence value), (4) $XU \rightarrow V$ does not hold ($XU \rightarrow V$ not frequent or $XU \rightarrow V$ has a low confidence value). Given a a belief $U \rightarrow V$, an unexpected rule $un.(X, Y)$ is modeled by the following n-ary query:

[1] The symbol $\sqcup$ denotes the disjoint union operator. It states that for a rule, patterns representing respectively premises and conclusion must be disjoint.

$$un.(X,Y) \equiv \begin{cases} freq(Y \cup V) = 0 \wedge \\ freq(X \cup U) \geq minfr_1 \wedge \\ freq(X \cup U \cup Y) \geq minfr_2 \wedge \\ freq(X \cup U \cup Y)/freq(X \cup U) \geq minconf \wedge \\ (freq(X \cup U \cup V) < maxfr \vee freq(X \cup U \cup V)/freq(X \cup U) < maxconf) \end{cases}$$

Classification Conflicts. Classification based on associations [11] is an other area where n-ary queries enable us to combine local patterns to help to design classifiers. Let C and C' be the items denoting the class values. The following example detects classification conflicts, here a pair of frequent classification rules $X \rightarrow C$ and $Y \rightarrow C'$ having confidences greater than a minimal threshold $minconf$. The rules have a large overlapping between their premises that may introduce classification conflicts on unseen examples.

$$classif.\ conflict(X,Y) \equiv \begin{cases} freq(X) \geq minfr \wedge \\ freq(Y) \geq minfr \wedge \\ freq(X \sqcup \{C\})/freq(X) \geq minconf \wedge \\ freq(Y \sqcup \{C'\})/freq(Y) \geq minconf \wedge \\ 2 \times length(X \cap Y) \geq (length(X) + length(Y))/2 \end{cases}$$

3.2 Soving n-ary Queries Using CP

After having formulated n-ary queries in a high level modeling as a set of numeric and set constraints as previously seen, these constraints are solved by a CP solver (the Gecode system[2]). Our method has 3 steps. Firstly, the dataset and the patterns involved in the n-ary query are linked. Then, unknown patterns are modeled using variables. Finally, numeric constraints and set constraints are reformulated in a low level way (for more details, see [5]). As the resolution performed by the CP solver is sound and complete, our approach is able to mine the correct and complete set of patterns satisfying n-ary queries.

4 Experiments

Experiments were performed on several datasets from the UCI repository[3] and a real-world dataset `Meningitis` coming from the Grenoble Central Hospital (329 transactions described by 84 items). Experiments were conducted with several kinds of n-ary queries: exception rules, unexpected rules and classification conflicts. We use a PC having a 2.83 GHz Intel Core 2 Duo processor and 4 GB of RAM, running Ubuntu Linux.

Highlighting Useful Patterns. Exception rules are a particular case of rare rules. Even when rare rules can be extracted [10], it is impossible to pick the exception rules among the set of all the rare rules. It is a pity because most of the rare rules are unreliable and it is much more interesting to get the exceptions rules. Fig. 1 quantifies the number of exception rules on the `Meningitis`

[2] `http://www.gecode.org`
[3] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

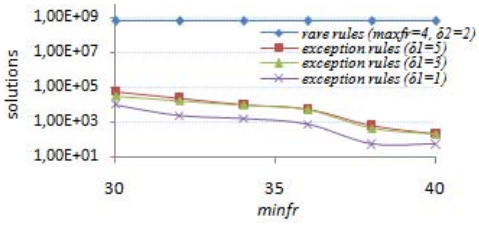


Fig. 1. Number of pairs of exception rules versus number of rare rules (`Meningitis`)

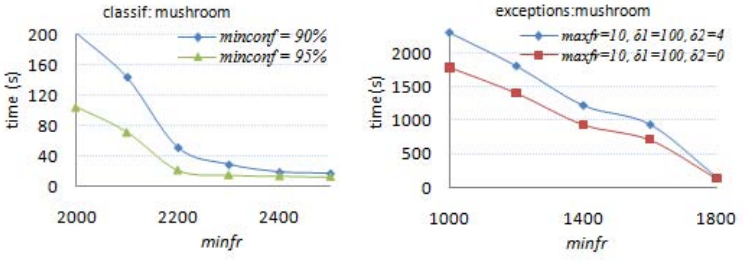


Fig. 2. Runtimes

dataset versus the number of rare rules (the number of rare rules depends on $maxfr$ and corresponds to the line at the top of the figure). Looking for exception rules reduces on several orders of magnitude the number of outputted patterns. Unexpected rules may also reveal useful information. For example, still on `Meningitis`, such a rule has a premise made of a high percentage of immature band cells and the absence of neurological deficiency and its conclusion is a normal value of the polynuclear neutrophil level. This rule is unexpected with the belief that high values of the white cells count and the polynuclear percentage lead to a bacterial etiological type.

Computational Efficiency. These experiments quantify runtimes and the scalability of our approach. Runtimes vary according to the size of the datasets but also the tightness of constraints[4]. On `Meningitis` and `Australian`, the set of all solutions is computed in a few seconds (less than one second in most of the cases). On `Mushroom`, runtimes vary from few seconds for tight constraints to about an hour for low frequency and confidence thresholds. These results suggest to conduct further experiments on this dataset to better evaluate the runtimes. Fig. 2 details the runtime of our method on `Mushroom` according to different thresholds of confidence and frequency. We observe that the tighter the

[4] A constraint is said *tight* if its number of solutions is low compared to the cardinality of the cartesian product of the variable domains, such as constraints defined by high frequency and confidence thresholds

constraint is, the smaller the runtime is. Indeed, tight constraints enable a better filtering of the domains and then a more efficient pruning of the search tree.

Obviously, our generic n-ary approach can be used for mining local patterns. We obtain on this task the same runtimes as [2] which were competitive with state of the art miners. With exception rules, we cannot compare runtimes because they are not indicated in [9].

5 Conclusion and Future Works

In this paper, we have presented a correct and complete approach to model and mine n-ary patterns. The examples described in Section 3.1 illustrate the generality and the flexibility of our approach. Experiments show its relevance and its feasibility in spite of its generic scope. For CSPs, all variables are existentially quantified. Further work is to introduce the universal quantification: this quantifier would be precious to model important queries such as the *peak* query[5].

References

1. Bonchi, F., Giannotti, F., Lucchese, C., Orlando, S., Perego, R., Trasarti, R.: A constraint-based querying system for exploratory pattern discovery. Inf. Syst. 34(1), 3–27 (2009)
2. De Raedt, L., Guns, T., Nijssen, S.: Constraint Programming for Itemset Mining. In: ACM SIGKDD Int. Conf. KDD 2008, Las Vegas, Nevada, USA (2008)
3. De Raedt, L., Zimmermann, A.: Constraint-based pattern set mining. In: 7th SIAM Int. Conf. on Data Mining. SIAM, Philadelphia (2007)
4. Khiari, M., Boizumault, P., Crémilleux, B.: Local constraint-based mining and set constraint programming for pattern discovery. In: From Local Patterns to Global Models (LeGo 2009), ECML/PKDD 2009 Workshop, Bled, Slovenia, pp. 61–76 (2009)
5. Khiari, M., Boizumault, P., Crémilleux, B.: Constraint programming for mining n-ary patterns. In: Cohen, D. (ed.) CP 2010. LNCS, vol. 6308, pp. 552–567. Springer, Heidelberg (2010)
6. Knobbe, A., Ho, E.: Pattern teams. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 577–584. Springer, Heidelberg (2006)
7. Ng, R.T., Lakshmanan, V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: Proceedings of ACM SIGMOD 1998, pp. 13–24. ACM Press, New York (1998)
8. Padmanabhan, B., Tuzhilin, A.: A belief-driven method for discovering unexpected patterns. In: KDD, pp. 94–100 (1998)
9. Suzuki, E.: Undirected Discovery of Interesting Exception Rules. Int. Journal of Pattern Recognition and Artificial Intelligence 16(8), 1065–1086 (2002)
10. Szathmary, L., Valtchev, P., Napoli, A.: Generating Rare Association Rules Using the Minimal Rare Itemsets Family. Int. J. of Software and Informatics 4(3), 219–238 (2010)
11. Yin, X., Han, J.: CPAR: classification based on predictive association rules. In: proceedings of the 2003 SIAM Int. Conf. on Data Mining, SDM 2003 (2003)

[5] The *peak* query compares neighbor patterns; a *peak* pattern is a pattern whose all neighbors have a value for a measure lower than a threshold

Neighborhood Based Clustering Method for Arbitrary Shaped Clusters

Bidyut Kr. Patra[1] and Sukumar Nandi[2]

[1] Department of Computer Science & Engineering, Tezpur University, Assam-784 028, India
[2] Department of Computer Science & Engineering, Indian Institute of Technology Guwahati, Guwahati, Assam-781039, India
{bidyut,sukumar}@iitg.ernet.in

Abstract. Discovering clusters of arbitrary shape with variable densities is an interesting challenge in many fields of science and technology. There are few clustering methods, which can detect clusters of arbitrary shape and different densities. However, these methods are very sensitive with parameter settings and are not scalable with large datasets. In this paper, we propose a clustering method, which detects clusters of arbitrary shapes, sizes and different densities. We introduce a parameter termed $Nearest\ Neighbor\ Factor\ (NNF)$ to determine relative position of an object in its neighborhood region. Based on relative position of a point, proposed method expands a cluster recursively or declares the point as outlier. Proposed method outperforms a classical method DBSCAN and recently proposed TI-k-Neighborhood-Index supported NBC method.

1 Introduction

Clustering problem appears in many different fields like data mining, pattern recognition, statistical data analysis, bio-informatics, *etc*. Clustering problem can be defined as follows. Let $\mathcal{D} = \{x_1, x_2, x_3, \ldots, x_n\}$ be a set of n patterns, where each x_i is N-dimensional vector in the given feature space. Clustering activity is to find groups of patterns, called clusters in $\mathcal{D}$, in such a way that patterns in a cluster are more similar to each other than patterns in distinct clusters. Many clustering methods have been developed over the years [1,2,3]. Clustering methods are mainly divided into two categories based on the criteria used to produce results *viz., distance based* and *density based*.

Distance based methods optimize a global criteria based on the distance between patterns. Clustering methods like k-means, CLARA, CLARANS are few examples of distance based method. Density based clustering methods optimize local criteria based on density distribution of the patterns.The advantage of density based clustering methods is that it can find clusters of arbitrary shapes and sizes in data.

DBSCAN (A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise) is a classical density based clustering method which can find clusters of arbitrary shapes and sizes in data [4]. However, DBSCAN is very sensitive to parameters (radius of neighborhood and minimum number of points in the neighborhood). It cannot find clusters of variable densities in data.

S.Zhou [5] proposed a neighborhood based clustering (NBC) method, which can discover clusters of arbitrary shape, size and variable densities. The core idea of NBC

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 306–315, 2011.

is Neighborhood Density Factor (NDF) which measures local density of a point. NDF of a point is ratio of the number of reverse K nearest neighbors to the number of K nearest neighbors. Based on value of NDF, a point is considered as a genuine cluster point (an inlier) or an outlier point. NBC proceeds by expanding a cluster from an arbitrary inlier point in the cluster. However, NBC does not consider relative position of a point in its neighborhood for computing NDF of the point. As a result, NBC may not detect border points correctly and the method is sensitive to K values. M.Kryszkiewicz et al. [6] showed that this method could not work well with high dimensional large datasets. They proposed an indexing method called TI-k-Neighborhood-Index to speed up the NBC method.

In this paper, we propose a clustering method referred to as *Nearest Neighbor Factor based Clustering (NNFC)*, which is less sensitive to parameter settings and suitable for large datasets. We introduce a parameter termed as *Nearest Neighbor Factor (NNF)* to determine relative position of an object in its neighborhood region. The *NNF* of a point with respect to a neighbor is the ratio of the distance between the point and the neighbor, and the average neighborhood distance of the neighbor. Unlike NDF, NNF accounts closeness (distance between a point and its neighbor) as well as neighborhood density of the neighbor. Based on average NNF value of a point, proposed method either expands a cluster recursively or declares it as outlierpoint. *NNFC* is tested with synthetic and real world datasets and it is found that *NNFC* outperforms TI-k-Neighborhood-Index supported NBC in clustering quality as well as execution time. Further, clustering produced by *NNFC* is close to natural clustering of dataset (as per Purity measure).

Rest of the paper is organized as follows. Section 2 describes proposed clustering method, with experimental results and discussion in section 3. Finally, section 4 concludes the paper.

2 Proposed Clustering Method

In this section, a formal definition for $Nearest\ Neighbor\ Factor\ (NNF)$ is given. Subsequently proposed clustering method *NNFC* is discussed in details.

NNFC uses K nearest neighbors statistics of each point to determine its position in its neighborhood region. To make this article more convenient to reader, we start with definition of K nearest neighbors , average K nearest neighbor distance and followed by $Nearest\ Neighbor\ Factor\ (NNF)$, key concept of this paper. Let $\mathcal{D}$ and d be a dataset and a distance function, respectively.

Definition 1 **($K\ Nearest\ Neighbor\ (KNN)\ Set$). :** *Let x be a point in $\mathcal{D}$. For a natural number K, a set $NN_K(x) = \{q \in \mathcal{D} | d(x,q) \leq d(x,q'), q' \in \mathcal{D}\}$ is called KNN of x if the following two conditions hold.*
(i) $|NN_K(x)| > K$ if q' is not unique in $\mathcal{D}$ or $|NN_K| = K$, otherwise.
(ii) $|NN_K(x) \setminus N^{(q')}| = K - 1$, where $N^{(q')}$ is the set of all q' point(s).

The $KNN\ Set(x) \subset \mathcal{D}$ is a set of K nearest points from x in $\mathcal{D}$. $KNN\ Set(x)$ includes all K^{th} nearest points if K^{th} nearest point is not unique.The $KNN\ Set$ of x is also called neighborhood of x. In this article, we use it interchangeably.

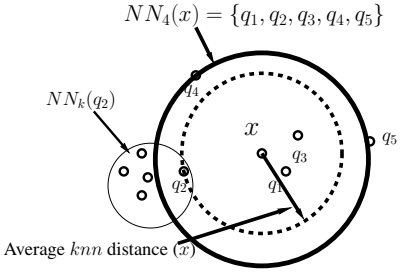


Fig. 1. The K nearest neighbor of x with $k = 4$

Definition 2 (*Average KNN distance*)**. :** *Let $NN_K(x)$ be the KNN set of a point $x \in \mathcal{D}$. Average KNN distance of x is average of distances between x and $q \in NN_K(x)$, i.e. Average KNN distance* $(x) = \dfrac{\sum_q d(x, q \mid q \in NN_K(x))}{|NN_K(x)|}$

Average KNN distance of a point x is average of distances between x and its K neighbors (Fig. 1). If *Average KNN distance* of x is less compared to other point y, it indicates that x's neighborhood region is more densed compared to the region where y resides.

Using information of K nearest neighbors and *Average KNN distance*, we define a new parameter called *Nearest Neighbor Factor* (NNF), which is the key idea of the proposed method.

Definition 3 (*Nearest Neighbor Factor* (NNF))**. :** *Let x be a point in $\mathcal{D}$ and $NN_K(x)$ be the KNN of x. The NNF of x with respect to $q \in NN_K(x)$ is the ratio of $d(x, q)$ and Average KNN distance of q.*

$$NNF(x, q) = d(x, q)/Average\ KNN\ distance(q)$$

The NNF of x with respect to a neighbor q is the ratio of distance between x and q, and *Average KNN distance* of q. $NNF(x, q)$ indicates x's relative position with respect to q's neighborhood. Based on values of NNF, our method $NNFC$ can identify regions with variable densities, outlier points and inlier points (cluster point). NNF values of a point $x \in \mathcal{D}$ w.r.t. to all its neighbors are close to 1 indicates that neighborhood regions of all neighbors belong to a cluster and x is an inlier point of the cluster. If NNF values of x varies significantly with respect to its neighbors, then neighborhood regions of x's neighbors may belong to different clusters. We consider average of NNF values of a point instead of individual values. Average of NNF values of x termed $NNF_{avg}(x)$ is calculated using Eq.(1).

$$NNF_{avg}(x) = \frac{\sum_q NNF(x, q \mid q \in NN_K(x))}{|NN_K(x)|} \tag{1}$$

The value of $NNF_{avg}(x)$ indicates x's relative position with respect to its neighbors. It can be shown that for uniformly distributed points, value of NNF_{avg} of each point

is 1. However, value of NNF_{avg} varies in real world datasets and it can be less than or more than 1 depending upon density of the region where point resides.

It is trivial to mention that $NNF_{avg}(x) = 1$ indicates that x and all its neighbors are in a cluster. Whereas, in case of $NNF_{avg}(x) \approx 1$, point x may club with a cluster of its neighbors. Otherwise, x can be noise point or another cluster point. Considering these points, we define following positions of a point $x \in \mathcal{D}$ with respect to a cluster. The parameter δ is used to define closeness of a point to a cluster.

Definition 4 (Border Point of a Cluster (BP)). : *Point x is called a border point of a cluster if $NNF_{avg}(x) < 1 - \delta$ and there exists a point y such that $1 - \delta \leq NNF_{avg}(y) \leq 1 + \delta$ and $y \in NN_K(x)$.*

Definition 5 (Core Point or Cluster Point (CP)). : *Point x is a Core Point or Cluster Point (CP) if $1 - \delta \leq NNF_{avg}(x) \leq 1 + \delta$.*

Core point resides inside in a cluster, whereas border point at boundary region of a cluster. A border point may be at the boundaries of more than one clusters. However, our method considers it in a cluster whose core point discovers it first.

Definition 6 (Outlier Point(OP)). : *Let x be a point in $\mathcal{D}$. The point x is a Outlier (OP) if $NNF_{avg}(x) > 1 + \delta$ and there is no CP in its KNN set.*

It may be noted that definition of outlier given here is in the line with Chandola et al. [7]. Using the above classifications of the data points in the similar line of DBSCAN, we establish following relationships between any two arbitrary points x, y in $\mathcal{D}$ and exploit them in the proposed method using only K neighborhood information.

Definition 7 (Directly Reachable). : *Let $NN_K(x)$ be KNN set of x. The point y is directly reachable from x if x is a CP and $y \in NN_K(x)$.*

In general, directly reachable is an asymmetric relation. However, if y is a core point, then directly reachable is symmetric one.

Definition 8 (Reachable). : *A point y is reachable from x if there is a sequence of points $p_1 = x, p_2, \ldots, p_m = y \in \mathcal{D}$ such that p_{i+1} is directly reachable from p_i, where $i = 1..(m-1)$*

Reachable is a transitive relation and it is a symmetric if y is a core point. Otherwise,it is asymmetric. Starting from a core point of a cluster, we can visit all points of a cluster using reachable relation. It may be noted that reachable relation is formed surrounding a core point. However, two border points of a cluster can also be related through a core point of the cluster. This relation called neighborhood connected is defined as follows.

Definition 9 (Neighborhood Connected). : *Two points x and y are called neighborhood connected if any one of the following conditions holds.*
(i) y is reachable from x or vice-versa.
(ii) x and y are reachable from a core point in the dataset.

Obviously, two core points in a cluster are neighborhood connected. Now, we can define Nearest Neighbor Factor based cluster and outlier points as follow.

Definition 10 (*$Neareat\ NeighborFactor\ based\ Cluster(NNFC)$*)**. :** *Let $\mathcal{D}$ be a dataset. $C \subset \mathcal{D}$ is a NNFC cluster such that*
(i) if $x, y \in C$, x and y are neighborhood connected and (ii) if $x \in C$, $y \in \mathcal{D}$ and y is reachable from x, then y also belongs to NNFC cluster C.

Definition 11 (NNFC based Outlier). : *Let $C_1, C_2, \ldots, C_m$ be NNFC clusters of dataset $\mathcal{D}$. The NNFC based outliers set is defined as $\mathcal{O} = \{p \in \mathcal{D} \mid p \notin C = \cup_{i=1}^{m} C_i\}$*

Algorithm 1. *NNFC($\mathcal{D}$, K,δ)*

```
/* D is a dataset, K is the value of K nearest neighbor, δ is variation parameter */
for each pattern x ∈ D do
   Calculate NN_K(x).
   Calculate NNF_avg(x).
end for
Each point is marked as 'undiscovered'.
cluster_id = 0;
for each pattern x ∈ D do
   if x is not marked as 'discovered' then
      Mark x as 'discovered'.
      if NNF_avg(x) < 1 − δ and NNF_avg(x) > 1 + δ then
         Mark x as 'noise'.
      else
         /*  x is a CP and start expanding a new cluster with id cluster_id + 1  */
         cluster_id = cluster_id + 1; QUEUE = ∅.
         for each p ∈ NN_K(x) do
            if p is 'undiscovered' or marked as 'noise' then
               Mark p with 'discovered' and cluster number cluster_id
            end if
            if p is a core point then
               QUEUE = QUEUE ∪ {p}.
            end if
         end for
         while QUEUE is not empty do
            Take a pattern y from QUEUE.
            for each q ∈ NN_K(y) do
               if q is 'undiscovered' or marked as 'noise' then
                  Mark q with 'discovered' and cluster number cluster_id
               end if
               if q is a core point then
                  QUEUE = QUEUE ∪ {q}.
               end if
            end for
            Remove y from QUEUE.
         end while
      end if
   end if
end for
```

Proposed clustering method, *NNFC* has two major steps as shown in Algorithm 1. In first step, it searches K nearest neighbors and computes NNF_{avg} of each pattern in the dataset. To speed up the searching process, we use TI-k-Neighborhood-Index [6] for finding K nearest neighbors. This indexing scheme avoids a large number of distance computations using triangle inequality property. Due to space limitation, we avoid detailed discussion of TI-k-Neighborhood-Index.

All points are marked as 'undiscovered' in the beginning of second phase. $NNFC$ picks an 'undiscovered' point (say, x) and marks x as 'noise' and 'discovered', if x is not a core point (CP). Otherwise, if x is a CP, it starts expanding a new cluster by finding all reachable points from x. A reachable point which is either 'undiscovered' or 'noise' is included into the new cluster. Each point in the cluster is marked with $cluster\_id + 1$ (initialized with $cluster\_id = 0$) and 'discovered'. It may be noted that a point earlier marked as 'noise' is included as a border point of the current cluster. Similarly, a reachable point which was earlier included as a border point of other cluster is not considered for the current cluster. This process continues until all points are marked as 'discovered'. Finally, points which are marked as 'noise' are declared as outlier set and points which are marked with $cluster\_id$ as cluster points. Needless to mention that *NNFC* finds arbitrary shaped clusters in the data.

We analyze time and space complexity of *NNFC*. Time complexity of the first step of *NNFC* is same as the complexity of TI-k-Neighborhood-Index, i.e. $O(n^2)$. In the second step, *NNFC* walks on each data points and expands a point if necessary. Time complexity of this step is $O(nK) = O(n)$. Overall time complexity of *NNFC* is $O(n^2)$. Space complexity is $O(nN)$, where N is the dimension of dataset.

3 Performance Evaluations

To evaluate effectiveness, *NNFC* is implemented using C language on Intel(R) Core 2 Duo CPU (3.0 GHz) Desktop PC with 4 GB RAM. The method is tested with two synthetic and two real world datasets. Comparisons with DBSCAN and TI-k-Neighbor-Index supported NBC [6] are reported in this section.

Rand-Index [8] and Purity [9] quality measures are used for comparison purpose. Let $\mathcal{P}$ be the set of all clusterings of $\mathcal{D}$. Rand Index (RI) is defined as follows.
$RI : \mathcal{P} \times \mathcal{P} \rightarrow [0, 1]$. For $\pi_1, \pi_2 \in \mathcal{P}$,

$$RI(\pi_1, \pi_2) = (a + e)/(a + b + c + e)$$

where, a is the number of pairs of patterns belonging to a same cluster in π_1 and to a same cluster in π_2, b is the number of pairs belonging to a same cluster in π_1 but to different clusters in π_2, c is number of pairs belonging to different clusters in π_1 but to a same cluster in π_2, and e is the number of pairs of patterns belonging to different clusters in π_1 and to different clusters in π_2. Similarly, *Purity* is defined as follows.

$$Purity(\pi_1, \pi_2) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\pi_2|} \max_j | \, C_i^{(2)} \cup C_j^{(1)} \, |, \text{where } C_i^{(2)} \in \pi_2, C_j^{(1)} \in \pi_1.$$

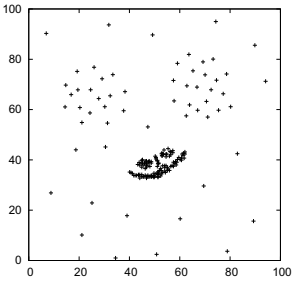
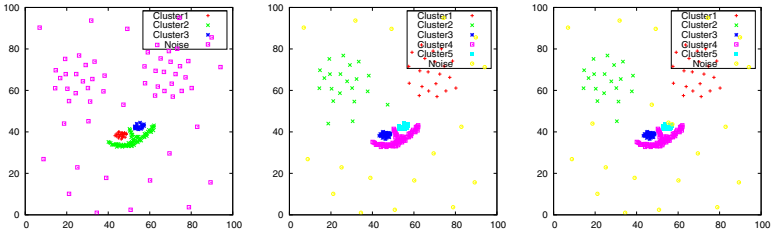


Fig. 2. Dataset [5]



(a) Clustering by DBSCAN (b) Clustering by by NNFC (c) Clustering by NBC

Fig. 3. Clustering results with Variable density dataset [5]

Detailed description of experimental results and datasets used are given below. In our experiments, we consider the value of $\delta = 0.25$. However, slight change in value of δ does not affect on clustering results.

3.1 Synthetic Datasets

Variable Density Dataset [5]: We use a synthetic dataset (Fig.2) which is similar to a dataset in [5]. This dataset consists of five clusters with variable densities. DBSCAN cannot find all five clusters with all possible values of parameters. With high density setting, DBSCAN can find only three clusters and treats other two as noise (Fig. 3(a)). However, *NNFC* finds all five clusters as does TI-k-Neighbor-Index supported NBC (Fig. 3(b), 3(c)).

Spiral Dataset: This dataset has two spiral (arbitrary) shaped clusters with 3356 patterns as shown in Fig. 4. From results of *NNFC* and TI-k-Neighbor-Indexed NBC with Spiral dataset, it is found that *NNFC* remains less sensitive to K values compared to TI-k-Neighbor-Indexed NBC method. Proposed method works well with a wide range of values ($K = 10$ to $K = 30$). It can detect two genuine clusters correctly with $K = 10$ to $K = 30$ (Fig. 5). With $K = 10$, TI-k-Neighbor-Index supported NBC finds four clusters splitting each of the genuine clusters into two (Fig. 6(a)). With $K = 15$, $K = 17$ and $K = 20$, it declares genuine cluster points as noise points

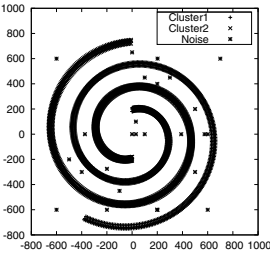


Fig. 4. Spiral Dataset

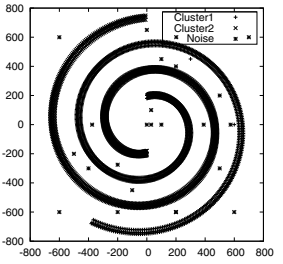
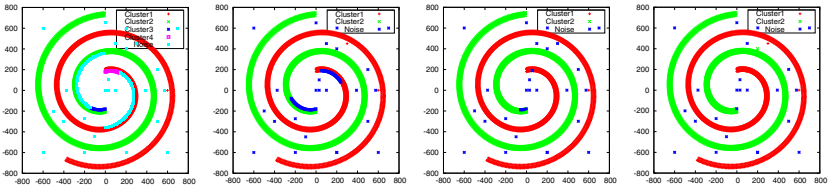


Fig. 5. Clustering by NNFC with $K = 10$ to $K = 30$



(a) K=10 (b) K=15 (c) K=17 (d) K=20

Fig. 6. Clustering By NBC/TI-k-Indexed NBC with different K values

(Fig 6(b), 6(c), 6(d)). Clustering results and execution time of *NNFC* and TI-k-Indexed NBC are reported with $\delta = 0.25$ in Table 1.

3.2 Real World Datasets

Shuttle Dataset: This dataset has 9 integer valued attributes of $58,000$ patterns distributed over 7 classes (after merging training and test sets). Class labels are eliminated from the all patterns. Clustering quality and time taken by the *NNFC* and TI-k-Index supported NBC method with different values of K are reported in Table 2. Execution time of TI-k-Index supported NBC method and *NNFC* methods are reported (Table 2). From results, it can be observed that Rand Index of *NNFC* method ($RI = 0.854$) is

Table 1. Experimental Results with Spiral Dataset

Value of K	Method	Time (in Seconds)	Purity	Rand Index	#Cluster
10	TI-k-Indexed NBC	0.91	0.951	0.899	4
	NNFC	**0.41**	**0.999**	**0.999**	**2**
15	TI-k-Indexed NBC	1.19	0.976	0.964	3
	NNFC	**0.43**	**0.999**	**0.999**	**2**
20	TI-k-Indexed NBC	1.48	0.998	0.998	3
	NNFC	**0.48**	**0.999**	**0.999**	**2**
25	TI-k-Indexed NBC	1.76	0.998	0.998	3
	NNFC	**0.55**	**0.998**	**0.998**	**2**

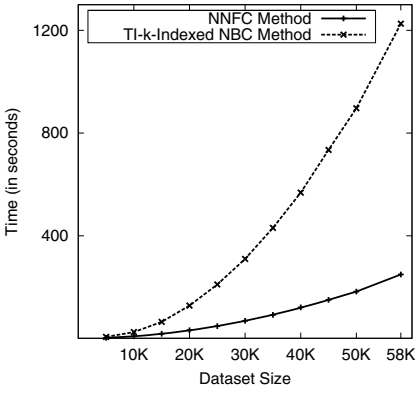


Fig. 7. Execution Time of NNFC and TI-k-Indexed NBC Methods with Shuttle Data

Table 2. Results with Real World Datasets (UCI Machine Learning Repository)

Dataset Used	Value of K	Method	Time (in Seconds)	Purity	Rand Index	Number of Clusters
Shuttle	30	TI-k-Indexed NBC	1121.94	0.913	0.585	28
		NNFC	**241.46**	**0.921**	**0.854**	**11**
	35	TI-k-Indexed NBC	1225.92	0.914	0.674	18
		NNFC	**245.76**	**0.921**	**0.855**	**9**
	40	TI-k-Indexed NBC	1290.71	0.914	0.799	15
		NNFC	**250.25**	**0.915**	**0.812**	**8**
	50	TI-k-Indexed NBC	1467.50	0.913	0.801	13
		NNFC	**265.10**	**0.918**	**0.844**	**7**
Page	15	TI-k-Indexed NBC	2.73	0.924	0.747	13
		NNFC	**0.77**	**0.931**	**0.831**	**6**
	25	TI-k-Indexed NBC	3.41	0.925	0.779	9
		NNFC	**0.88**	**0.930**	**0.850**	**9**

better than the TI-k-Index supported NBC method ($RI = 0.585$) for the value of $K = 30$. The number of clusters produced by *NNFC* is consistently better than TI-k-Index supported NBC method with different values of $K = 30, 35, 40, 50$ and number of clusters (8) produced by *NNFC* is close to the number of actual clusters (7) of the dataset for $K = 50$. Execution time of *NNFC* is more than five times faster than TI-k-Index supported NBC method for Shuttle dataset.

Fig. 7 shows execution time of the *NNFC* and TI-k-Index supported NBC methods as data size varies from 5000 to 58000 for Shuttle dataset with $\delta = 0.25$. It may be noted that $NNFC$ consumes significantly less time as dataset size increases.

Page: This dataset has 5473 patterns distributed across five classes. A pattern corresponds to a page block of 54 documents. From experimental results, it can be noted that *NNFC* produces better clustering (Purity$=0.931, 0.930$ and Rand Index$=0.831, 0.850$)

than TI-k-Index supported NBC method (Purity $= 0.924, 0.925$ and Rand Index$=$ $0.747, 0.779$). The *NNFC* takes less time compared to TI-k-Index supported NBC method (Table 2). With $K = 25$, both the methods produce equal number of clusters (10) for this dataset. However, clustering result ($RI = 0.850$) of *NNFC* is better than result ($RI = 0.779$) produced by the TI-k-Index supported NBC.

4 Conclusions

NNFC is a neighborhood based clustering method, which can find arbitrary shaped clusters in a dataset with variable densities. The $NNFC$ calculates $Nearest\ Neighbor\ Factor$ to locate relative position of a point in dataset. Based on position, a point is declared as a cluster point or noise point. Experimental results demonstrate that it outperforms TI-k-Index supported NBC in both clustering results and execution time.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley Interscience Publication, New York (2000)
2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
3. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks **16**(3) (May 2005) 645–678
4. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings of 2nd ACM SIGKDD, pp. 226–231 (1996)
5. Zhou, S., Zhao, Y., Guan, J., Huang, J.Z.: A neighborhood-based clustering algorithm. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 361–371. Springer, Heidelberg (2005)
6. Kryszkiewicz, M., Lasek, P.: A neighborhood-based clustering by means of the triangle inequality. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 284–291. Springer, Heidelberg (2010)
7. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Survey 41(3) (2009)
8. Rand, W.M.: Objective Criteria for Evaluation of Clustering Methods. Journal of American Statistical Association 66(336), 846–850 (1971)
9. Zhao, Y., Karypis, G.: Criterion functions for document clustering: Experiments and analysis. Technical report, University of Minnesota (2002)

FAST Sequence Mining Based on Sparse Id-Lists

Eliana Salvemini[1], Fabio Fumarola[1], Donato Malerba[1], and Jiawei Han[2]

[1] Computer Science Dept., Univ. of Bari, E. Orabona, 4 - 70125 Bari, Italy
{esalvemini,ffumarola,malerba}@di.uniba.it
[2] Computer Science Dept., Univ. of Illinois at Urbana-Champaign, 201 N Goodwin Avenue Urbana, IL 61801, USA
hanj@cs.uiuc.edu

Abstract. Sequential pattern mining is an important data mining task with applications in basket analysis, world wide web, medicine and telecommunication. This task is challenging because sequence databases are usually large with many and long sequences and the number of possible sequential patterns to mine can be exponential. We proposed a new sequential pattern mining algorithm called FAST which employs a representation of the dataset with indexed sparse id-lists to fast counting the support of sequential patterns. We also use a lexicographic tree to improve the efficiency of candidates generation. FAST mines the complete set of patterns by greatly reducing the effort for support counting and candidate sequences generation. Experimental results on artificial and real data show that our method outperforms existing methods in literature up to an order of magnitude or two for large datasets.

Keywords: Data Mining, Sequential Pattern Discovery, Sparse Id-List.

1 Introduction

Finding sequential patterns in large transactional databases is an important data mining task which has applications in many different areas. Many sequential pattern mining algorithms have been proposed in literature [1,2,4,5,9,3,6]. The main limits of current algorithms are: 1) the need of multiple scans of the database; 2) the generation of a potentially huge set of candidate sequences; 3) the inefficiency in handling very long sequential patterns.

The most demanding phases of a sequence mining process are candidate generation and support counting. Candidate generation is usually based on the *Apriori property* [1] according to which super-patterns of an infrequent pattern cannot be frequent. This property helps to prune candidates and to reduce the search space. Basing on this heuristic GSP [2] and SPADE [4] methods have been proposed. Since the set of candidate sequences includes all the possible permutations of the frequent elements and repetition of items in a sequence, A-priori based methods still generate a really large set of candidate sequences. The support counting phase requires multiple database scans each time the candidate sequence length grows, which is prohibitively expensive for very long sequences.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 316–325, 2011.

To deal with these issues, Pei *et al.* developed [3] a new approach for efficient mining of sequential pattern in large sequence database. *Prefixspan* adopts a depth-first search and a pattern growth principle. The general idea is to consider only the sequences prefix and project their corresponding postfix into projected databases. This way the search space is reduced in each step because projected databases are smaller than the original database. Sequential patterns are grown by exploring only the items local to each projected database. The drawback of Prefixspan is that for long sequences it needs to do many database projections.

Lately in 2002 Jay Ayres *et al.* present a faster algorithm, SPAM [5] which outperforms the previous works up to an order of magnitude for large databases. SPAM combines an efficient support counting mechanism and a new search strategy to generate candidates. The support counting is based on a data vertical representation in bitmaps and it does not require database scans. The drawback of SPAM is that it uses too much memory to store the bitmaps.

Another algorithm based on bitmap representation is HVSM [6]. HVSM is slightly faster than SPAM because it uses a different search strategy. At the beginning it generates all the frequent items or itemsets which constitute the first tree level. On these nodes it performs only sequence-extensions to generate sequences, by taking brother nodes as child nodes. This way HSVM is faster than SPAM because it discovers all the frequent itemsets at the first step.

To mine large databases of long sequences, we propose a new method FAST (Fast sequence mining Algorithm based on Sparse id-lisTs), which prevents multiple database scans to compute pattern support by indexing the original set of sequences with specific data structures called *sparse id-lists*. The database is read once and loaded in the main memory in an indexed form, such that each item is associated with the lists of all the sequence ids in which it appears in the database, preserving the transaction ordering. The support is directly computed from the data structures associated to each sequence without requiring database scans. Our proposed method uses the same lexicographical tree of HSVM so it gains advantage from the idea to mine all the frequent itemsets as first step.

2 Problem Definition

The problem of mining sequential patterns can be stated as follow. Let $I = \{i_1, i_2, ..., i_n\}$ be a set of different **items**. An **itemset** is a non-empty unordered subset of items (or a single item) denoted as $(i_1, i_2, ..., i_k)$ where i_j is an item for $1 < j < k$. A **sequence** is an ordered list of itemsets denoted as $\langle s_1 \rightarrow s_2 \rightarrow ... \rightarrow s_m \rangle$ where s_i is an itemset i.e. $s_i \subset I$ for $1 < j < m$. An item can occur at most once in an sequence itemset, but multiple times in different itemsets of a sequence. The number of items in a sequence is called the **length** of the sequence. A sequence of length k is called a **k-sequence**. A sequence $\alpha = \langle a_1 \rightarrow a_2 \rightarrow ... \rightarrow a_n \rangle$ is said a **subsequence** of another sequence $\beta = \langle b_1 \rightarrow b_2 \rightarrow ... \rightarrow b_m \rangle$ and β a **super sequence** of α, denoted as $\alpha \ll \beta$, if there exist integers $1 \leq j_1 \leq j_2 \leq ... \leq j_n \leq m$ such that $a_1 \subset b_{j1}$, $a_2 \subset b_{j2}$ and $a_n \subset b_{jn}$, *i.e.*, the sequence $A \rightarrow BC$ is a subsequence of the sequence $AB \rightarrow E \rightarrow BCD$,

while $AC \rightarrow AD$ is not a subsequence of $A \rightarrow C \rightarrow AD$. A **sequence database** D is a set of tuples $\langle sid, S\rangle$, where sid is a sequence id and S is a sequence. A **transaction** is formally an itemset. Each sequence has associated a list of transactions, each with a time stamp. We assume that no sequence has more than one transaction with the same time stamp, so that the transaction time can be the transaction id and we can also sort the transactions in the sequence according to the transaction time. A tuple $\langle sid, S\rangle$ is said to **contain** a sequence α if α is a subsequence of S, i.e. $\alpha \subset S$. The **support** or **frequency** of a sequence α in a database D, denoted as $\sigma(\alpha)$, is the number of tuples which contain α, i.e. $\sigma(\alpha) = \{\langle\ sid,\ S\ \rangle \mid (\langle\ sid,\ S\ \rangle\ \epsilon\ D) \wedge (\alpha \subset S)\}$. Given a user defined threshold δ called **minimum support**, denoted **min_sup**, a sequence α is said **frequent** in a database D if at least δ sequences in D contain α, i.e. $\sigma(\alpha) \geq \delta$. A frequent sequence is also called **sequential pattern**. A sequential pattern of length k is called an **k-pattern**. Given a sequence database D and a *min_sup* threshold δ, the problem of sequential pattern mining consist in finding the complete set of sequential patterns in the database D.

Example 1: Let D in Table 1 be a sequence database. Fixed $\delta = 2$. We omit the arrows in the sequences for brevity. The sequence $a(abc)(ac)d(cf)$ has 5 itemsets (transactions): a,(abc),(ac),d,(cf). Its length is 9, item a appears three times in the sequence so it contribute 3 to its length but the whole sequence contributes only 1 to the support of a. Sequence $\alpha = a(bc)$ is a subsequence of the sequences $a(abc)(ac)d(cf)$ and $(ad)c(bc)(ae)$, so $\sigma(\alpha) = 2$ and α is a sequential pattern.

Table 1. An example of sequence database D

Sequence_ID	Sequence
10	a (abc) (ac) d (cf)
20	(ad) c (bc) (ae)
30	(ef) (ab) (df) c b
40	e g (af) c b c

3 Algorithm

In this section we describe the algorithm, the lexicographic tree and the new data structure called *sparse id-list* upon which our algorithm is based.

Lexicographic Tree: Suppose we have a lexicographic ordering $\leq$ on the items I in the database. This ordering can be extended to sequences, by defining $a \leq b$ if sequence a is a subsequence of b. All sequences can be arranged in a lexicographic tree in which a node labeled with a is left brother of a node labeled with b if item a appears before item b in the lexicographic ordering. A node a is father of a node b if sequence in node a is a subsequence of sequence in node b.

A sequence can be extended with either a *sequence/itemset-extension*. A sequence-extended sequence is a sequence generated by adding a new transaction of a single item to the end of the sequence. An itemset-extended sequence is generated by adding an item to the last itemset of the sequence.

First Step: Itemset Extension. With a first database scan we compute the set I of all the frequent 1-items. Then we build a lexicographic *itemset tree* to compute all the frequent itemsets starting from I. The itemset tree has as root's children all the frequent 1-items and on these nodes we perform only *itemset-extensions*. Each node is iteratively expanded by taking its right brother nodes as child nodes. If the itemset obtained concatenating the labels of the two nodes is frequent, the candidate child node is added in the tree, otherwise it is pruned and that path no more expanded. After the *itemset-extension* all and only the frequent items and itemsets are contained in the lexicographic itemset tree. These are used as input to generate another lexicographic tree, the *sequence tree*, to mine all the frequent sequences.

Second Step: Sequence Extension. The first level of the *sequence tree* contains all the frequent items and itemsets generated in the first step. For sequences discovery, each node is iteratively expanded by taking all its brother nodes as child nodes, included itself. If the obtained sequence is frequent, then the new candidate child node is added in the tree, otherwise it is pruned and that path is not expanded. At the end of this process, the lexicographic *sequence tree* will contain all the sequential patterns.

4 Data Structure: Sparse Id-List

The support counting method of FAST is based on a novel data structure that we called *sparse id-list* (SIL). The concept of id-list is not new in literature, it was first introduced by SPADE [4]. SPADE id-list is a list of all the customer-id and transaction-id pair containing the element in the database. Its size is variable. Itemset/sequence-extensions are executed by joining the id-lists. This operation for long id-list is very expensive in time. Differently from SPADE, our *SIL* gives a horizontal representation of the database.

Definition 1. *Let D be a sequence database, $|D|$ the size of the D, $j \in (0,\ldots,|D|)$ be a sequence, as defined in section 2. For each frequent item/itemset i a* sparse id-lists *SIL_i can be defined as a vector of lists, where:*

- *the size of the vector of SIL_i is $|D|$,*
- *$SIL_i[j]$ is a list and contains the occurrences of i in the sequence j,*
- *if sequence j does not contains i, its list in position $SIL_i[j]$ has value null.*

The vector of SIL_i has as many rows as are the database sequences, its size is fixed and depends on $|D|$. Each position of the vector corresponds to a database sequence, and is a list which contains all the transaction ids of the sequence. We associate a *sparse id-list* to each item i in the sense that in the list of the *j-th* sequence there will be only the ids of transactions which contain item i in

sequence j, therefore the list size is not fixed. In the case an item appears in all the transactions of sequence j, the *j-th* list size will be equal to the *j-th* sequence length in the database, but in general it is shorter, that's why we named this data structure *sparse* id-lists. If an item does not appears in any transaction of a sequence, the list corresponding to that sequence in the vector will be *null*.

4.1 Support Counting for Item and Itemset Based on Sparse Id-List

A *SIL* is associated to each frequent item and itemset in the database and stored in memory. *SILs* for 1-items are incrementally built during the first database scan. *SILs* for itemsets are built during the itemset-estension from *SILs* of single items verifying the condition that the single items of itemset appear in the same transaction of the same sequence. The support is efficiently computed by just counting the number of lists in the *SIL* which are not *null*. This is done during the construction of the *SILs* itself without requiring a scan of the whole *SILs*.

Example 2: Consider the sequence database in Table 1 with $\sigma = 2$. With the first database scan FAST builds the *SILs* (Fig. 1(a)...2(a)) for all the 1-items, computes the supports and prunes all the *SILs* of unfrequent items. Item g is infrequent so its *SIL* has been removed. Now we perform an itemset-extension and build the *SIL* for itemset (ab) (Fig. 2(b)). To do this we have to scan all the rows in the *SILs* of a (Fig. 1(a)) and b (Fig. 1(b)) and for each row j we have to scan the list in it and report in the new *SIL* only the transaction ids which are equal between a and b. In this case at the first row the first transaction-id which is equal between a and b is 2, in the second row there isn't so we store *null*, at the third row it is 2, while in the fourth there isn't. *SIL* for itemset (ab) is now stored in memory along with *SILs* of frequent 1-items and it will be used in itemset-extension to compute the support for longer itemset ((abc) i.e.).

4.2 Support Counting for Sequence Based on Sparse Id-List

The support for sequences is computed from a derivative of *SIL* that we called *vertical-id-list* (*VIL*). It is a vector having the same size of the *SIL*. Each position of the vector refers to a particular sequence in the database, as for the *SIL*,

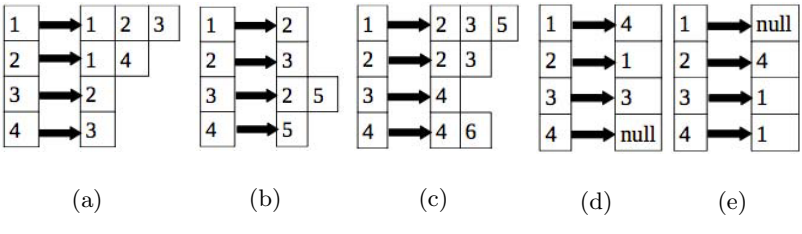


Fig. 1. From left to right, the *sparse id-lists* for items: a support $\sigma = 4$; b support $\sigma = 4$; c support $\sigma = 4$; d support $\sigma = 3$; e support $\sigma = 3$; f support $\sigma = 3$

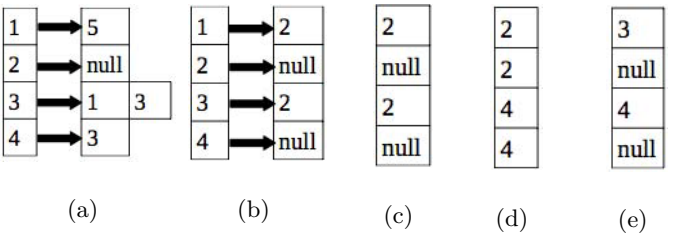


Fig. 2. From left to right: *sparse id-list* for itemset (ab) with support $\sigma(ab) = 2$; *vertical id-list* for itemset (ab), item c and sequence $(ab) \rightarrow c$

and stores only the first transaction id in which the element we are trying to concatenate in the sequence occurs, plus a reference to the *SIL* in which that transaction id is stored. During the sequence extension FAST stores in each node of the first level of the sequence tree a *VIL* for all the frequent items and itemsets computed in the first step. To generate the sequence $a \rightarrow b$ a new *VIL* is generated, verifying the condition that item b appears in the same sequence but in a transaction after items a. If this condition is not true we use the reference at the *SIL* stored in the *VIL* of b to move to the next transaction id greater than the transaction id of a. This transaction id will be stored in the new *VIL* of sequence $a \rightarrow b$ or *null* if it does not exist. The support of the sequence corresponds to the number of elements different than null in the *VIL* and, as for the *SIL*, it is computed during the construction of the *VIL* without requiring its scan. If the sequence $a \rightarrow b$ is frequent its *VIL* will be stored in the node $a \rightarrow b$ in the *sequence tree* and it will be used in the next sequence extension to compute the support of the super sequences of $a \rightarrow b$. The reader can notice that we only compute one scan of the database at the beginning of the algorithm to generate all the frequent 1-items and their corresponding *SILs*. Then the support for itemsets and sequences is directly computing by just counting the number of rows different than null in the *SIL* or *VIL*.

Example 3: Let us consider *SIL* for itemset (ab) in Fig. 2(b) and for item c in Fig. 1(c). Suppose we want to compute sequence $(ab) \rightarrow c$. We first have to get the *VILs* for elements (ab) and c from their *SILs* and stored in the corresponding nodes in the sequence tree. The *VILs* are showed in Figs. 2(c) and 2(d). For each position of the *VIL* of c, we search for a transaction id which is greater than the transaction id of the *VIL* of (ab) in the same position. If this condition is false we move to the next element using the reference at the *SIL* stored in the *VIL*. In this case the first transaction id in the *VIL* of c is 2 which is not greater than transaction id in the *VIL* of (ab), so we move to the next transaction id in the *SIL* of c (Fig. 1(c)) which is 3. The second transaction id is again 2 but since the transaction id in (ab) is null we have to skip to the following position. The next transaction id is 4 which is greater than the transaction id in (ab) 2 so it is stored in the new *VIL*. Finally the last transaction-id is 4 but since the

transaction id in (ab) is null the condition is not verified. The *VIL* for sequence $(ab) \rightarrow c$ is showed in Fig. 2(e).

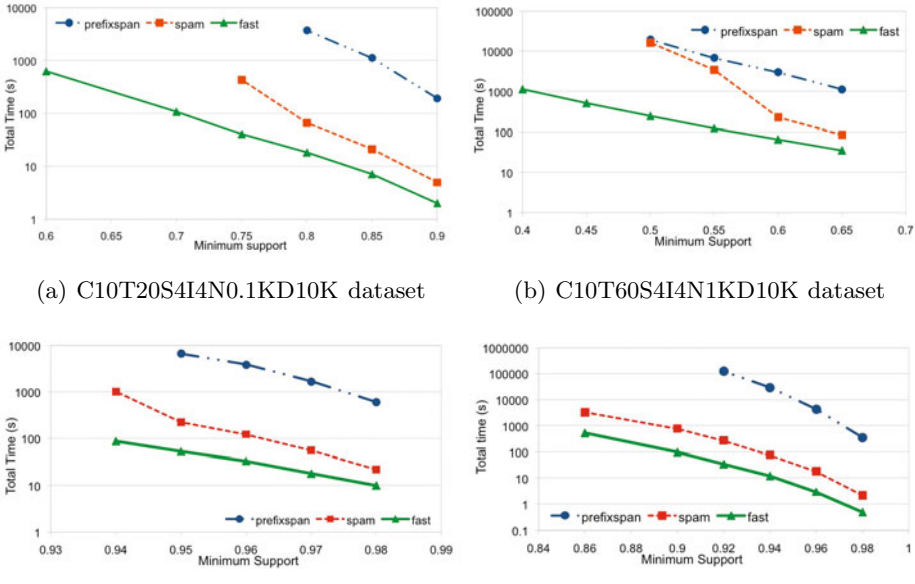
5 Experiments

We report the performances of FAST against PrefixSpan [3] and SPAM [5] by varying different database parameters. We did not compare FAST against SPADE [4] because authors in [5] demonstrate that SPAM always outperforms SPADE and against HSVM because the differences of performance between SPAM and HSVM are minimal as shown in [6]. We implemented FAST in java. The java codes/executables for PrefixSpan and SPAM were obtained from Philippe Fournier-Viger's [11] and Joshua Ho's websites [8]. The experiments were performed on a 2.4GHz Intel Xeon server with 10 GBs of RAM, running Linux.

Datasets: The synthetic datasets we used were generated using the IBM data generator [1], that mimics real-world transactions where people buy sets of items. We compared the performances by varying several factors which can be specified as input parameters to the data generator, with different minimum support values on small and large databases. These parameters are: D the number of customers in the dataset; C the average number of transactions per customer; T the average number of items (transaction size) per transaction; S the average size of maximal potentially large sequences; I the average size of itemset in maximal potentially large sequences and N the number of different items. We also compared the algorithms on a real dataset Pumsb-star, taken from the FIMI repository [10] which is dense and has 49046 long sequences of single items.

5.1 Performance Comparison

Figures 3(a)- 3(b) show performance comparison on small datasets. The datasets are identical, except that 3(a) has 100 distinct items (N=0.1k), whereas 3(b) has 1000 (N=1k). Figures 3(c), 3(d) and 4(a) show performance comparison on large datasets, with 100000 sequences (D=100k) with longer sequences (C=20, 35, 40). Experimental results shows that FAST has the best performance. It outperforms SPAM of one order of magnitude in the small and large datasets, and PrefixSpan till to two orders of magnitude on large datasets. For the support values where SPAM can run, it is generally in the second spot. It fails to run for lower support values because it needs huge memory to store the bitmaps. In all the experiments PrefixSpan turns out to be very slow compared to the other two algorithms. For long sequences it fails because it needs to do too many database projections. FAST instead can run faster also in presence of longer sequences with large datasets. Finally, Fig. 4(b) shows the performance comparison on Pumsb-star real dataset and on this dataset FAST is an order of magnitude faster than SPAM and two orders faster than PrefixSpan.



(a) C10T20S4I4N0.1KD10K dataset

(b) C10T60S4I4N1KD10K dataset

(c) C20T20S20I10N0.1KD100K dataset

(d) C40T40S20I20N1KD1K dataset

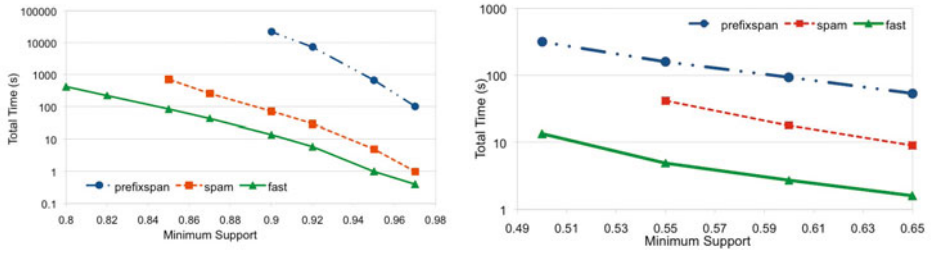
Fig. 3. Performance comparison varying minimum support on different datasets

To show that experimental results are not dependent on the language implementation, we compare our java version of FAST with the original C++ implementation of PrefixSpan [12]. Results in Fig. 5 show that FAST is still faster than PrefixSpan and starting from support 0.92 it is an order of magnitude faster than PrefixSpan.

5.2 Memory Usage

In Fig. 6 we presented graphics for the average memory consumption of the three algorithms on other synthetic datasets. We can quickly see as SPAM requires really much memory compared to FAST and PrefixSpan, up to one order of magnitude. SPAM is inefficient in space because even when an item is not present in a transaction it needs to store a zero in the bitmap to represent this fact. The trend of SPAM and FAST is comparable even if PrefixSpan requires less memory than FAST, because at each run it reduces the database size of one length avoiding to store sequence prefix in the database. FAST requires more memory than PrefixSpan because it needs to store in memory the *sparse id-lists* for all the frequent 1-items and itemsets. However, the memory usage of FAST is comparable to the one of PrefixSpan.

In Fig. 6(b) we present another kind of graphic which shows the total memory consumption of the three algorithms during the whole mining process, on the



(a) C35T40S20I20N1KD1K (b) pumsb-star

Fig. 4. Performance comparison on C35T40S20I20N1KD1K (a) and Pumsb-star (b)

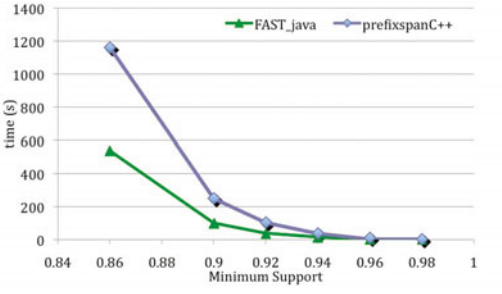
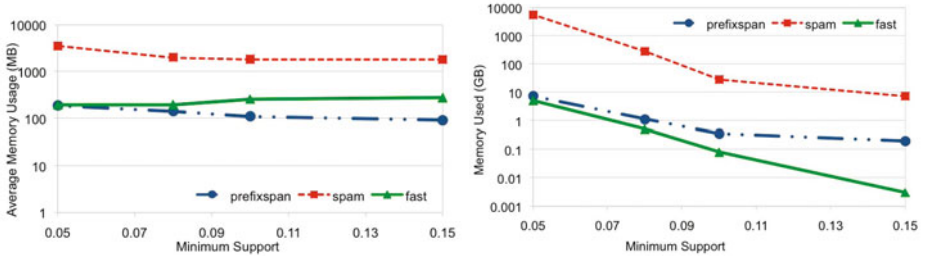


Fig. 5. Fast in Java vs. PrefixSpan in C++ on dataset C40T40S20I20N1KD1K



(a) D=5 C=15 T=10 S=10 I=10 (b) D=5 C=15 T=10 S=10 I=10

Fig. 6. Memory comparison on medium datasets

same datasets. It is interesting to notice that the total memory used by FAST is lower than the one used by SPAM and Prefixspan. This happens because FAST is faster than the other two algorithms in all the experiments that we performed and in this way it occupies less memory during the whole computation. We believe that even if the average memory consumption of FAST is slightly upper than PrefixSpan, the memory consumption of FAST is not a big issue as for

example in SPAM because the total memory that FAST uses is comparable and in some cases less than PrefixSpan.

6 Conclusions

We presented a new sequence mining algorithm FAST that quickly mines the complete set of patterns in a sequence database, greatly reducing the effort for support counting and candidate sequences generation phases. It employs a new data representation of the dataset based on sparse id-lists and indexed vertical id-lists, which allows to quickly access an element and count its support without database scans. Future work will consist of mining all the closed and maximal frequent sequences, as well as pushing constraints within the mining process to make the method suitable for domain specific sequence mining task.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: International Conference on Data Engineering (ICDE 1995), Taipei,Taiwan, vol. 0, pp. 3–14 (1995)
2. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Apers, P., Bouzeghoub, M., Gardarin, G. (eds.) Advances in Database Technology EDBT 1996, vol. 1057, pp. 1–17. Springer, Heidelberg (2006)
3. Pei, J., Han, J., Asl, M.B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: PrefixSpan Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth. In: Proc.17th Int'l Conf. on Data Eng., ICDE 2001, Heidelberg, Germany, pp. 215–226 (2001)
4. Zaki, M.J.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning 42, 31–60 (2001)
5. Ayres, J., Gehrke, J., Yiu, T., Flannick, J.: Sequential PAttern Mining using A Bitmap Representation, pp. 429–435. ACM Press, New York (2002)
6. Song, S., Hu, H., Jin, S.: HVSM: A New Sequential Pattern Mining Algorithm Using Bitmap Representation. In: Li, X., Wang, S., Dong, Z.Y. (eds.) Advanced Data Mining and Applications, vol. 3584, pp. 455–463. Springer, Heidelberg (2005)
7. Fournier-Viger, P., Nkambou, R., Nguifo, E.: A Knowledge Discovery Framework for Learning Task Models from User Interactions in Intelligent Tutoring Systems. In: Gelbukh, A., Morales, E. (eds.) MICAI 2008. LNCS (LNAI), vol. 5317, pp. 765–778. Springer, Heidelberg (2008)
8. Ho, J., Lukov, L., Chawla, S.: Sequential Pattern Mining with Constraints on Large Protein Databases (2008), http://sydney.edu.au/engineering/it/~joshua/pexspam/
9. Han, J., Pei, J., Asl, B.M., Chen, Q., Dayal, U., Hsu, M.C.: FreeSpan: frequent pattern-projected sequential pattern mining. In: KDD 2000: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data mining, Boston, MA, pp. 355–359 (2000)
10. Workshop on frequent itemset mining implementations FIMI 2003 in conjunction with ICDM 2003, http://fimi.cs.helsinki.fi
11. SPMF, http://www.philippe-fournier-viger.com/spmf/index.php
12. Illimine, http://illimine.cs.uiuc.edu/

From Connected Frequent Graphs to Unconnected Frequent Graphs*

Łukasz Skonieczny

Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
L.Skonieczny@ii.pw.edu.pl

Abstract. We present the UFC (Unconnected From Connected) algorithm which discovers both connected and disconnected frequent graphs. It discovers connected frequent graphs by means of any existing graph mining algorithm and then joins these graphs with each other creating unconnected frequent graphs with increasing number of connected components. We compare our method with previously proposed UGM algorithm and a *gSpan* variation.

Keywords: graph mining, unconnected frequent graphs.

1 Introduction

The purpose of frequent graphs mining is to find all graphs which are subgraph isomorphic with large number of graphs in the given database. In recent years several algorithms were proposed including gSpan [11], MoFa [2], Gaston [8], FFSM [3], SPIN [4], FSG [7], AGM [5], AcGM [6]. Most of algorithms discover only connected frequent graphs. Unconnected frequent graphs are sometimes more useful than connected ones as was shown in [10] on the example of contrast patterns. In [9] we proposed the UGM algorithm which discovers both connected and unconnected frequent graphs in one phase and experimentally showed that it was orders of magnitude faster than two-phase methods which depend on preliminary frequent connected graphs discovery. Further research showed that two-phase methods are still feasible and lead to algorithm which outperforms UGM. We propose the UFC algorithm (Unconnected From Connected) which uses any algorithm to discover connected frequent graphs and then joins them pairwise creating unconnected candidates. Filtering these candidates with frequent graphs lattice and using optimizations proposed for UGM makes the UFC faster than the UGM.

* This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: Interdisciplinary System for Interactive Scientific and Scientific-Technical Information.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 326–335, 2011.

2 Basic Notions

Definition 1. *(graph)* Undirected labeled graph G (called graph later on) is a tuple $G=(V, E, lbl, L)$ where V is a set of vertices, $E=\{\{v_1, v_2\} : v_1 \neq v_2 \in V\}$ is a set of edges, $lbl : V \cup E \rightarrow L$ is a labeling function, L is a set of labels.

Definition 2. *(connected and unconnected graph)* Graph is *connected*, if there is a path between every pair of its vertices. Otherwise, graph is *unconnected*.

Definition 3. *(graph isomorphism)* $G=(V, E, lbl, L)$ and $G'=(V', E', lbl', L')$ are isomorphic iff there exists bijective function $\phi : V \rightarrow V'$ such that:
$\forall e = \{v_1, v_2\} \in E : \{\phi(v_1), \phi(v_2)\} \in E'$ *and*
$\forall v \in V : lbl(v) = lbl'(\phi(v))$ *and*
$\forall e \in E : lbl(e) = lbl'(\phi(e))$.

Definition 4. *(subgraph and supergraph)* Graph $G' = (V', E', lbl', L')$ is a subgraph of $G = (V, E, lbl, L)$ (G is a supergraph of G') iff
$V' \subseteq V$ *and*
$E' \subseteq E$ *and*
$L' \subseteq L$ *and*
$\forall x \in V' \cup E' : lbl'(x) = lbl(x)$.

Definition 5. *(connected component)* Graph CG is a *connected component* of graph G, if CG is connected and CG is a maximal subgraph of graph G.

Definition 6. *(subgraph isomorphism)* Graph G' is subgraph isomorphic to graph G if it is isomorphic to some subgraph of G.

Definition 7. *(support)* Let D be a set of graphs. Support of a graph G, $sup(G)$, is defined as the number of graphs in D which G is subgraph isomorphic with.

Definition 8. *(frequent graph)* The graph G is called frequent in a set D if its support is greater than or equal to some user-defined threshold *minSup*.

Definition 9. *(edge descriptor) Edge descriptor* of the edge $e = \{v_1, v_2\} \in E$ is a pair $(\{lbl(v_1), lbl(v_2)\}, lbl(e))$.

Definition 10. *(edge set) Edge set* of G, $ES(G)$, is multiset of edge descriptors of edges from G. If G' is subgraph isomorphic to G then $ES(G') \subseteq ES(G)$.

3 From Connected to Unconnected

The proposed algorithm is based on the following property.

Property 1. If unconnected graph is frequent then each of its connected components is also frequent.

This property comes from the fact that each subgraph of the frequent graph is frequent as well. The core idea of UFC is to first find frequent connected graphs and then use them to generate unconnected candidates. The proposed method is similar to *Apriori* [1] algorithm, that is generating two-components candidates

form one-components frequent graphs, generating three-components candidates from two-components frequent graphs and so on. We use the following notation.

F_k	list of k-components frequent graphs,
K_k	list of k-components candidates,
$G = (C_1, C_2, \ldots C_k)$	k-components graph,
	C_i is a $i - th$ component of the G graph

We will also use small letters $a, b, c, \ldots$ to denote connected components and strings of these letters to denote unconnected graphs containing these components. Therefore $aabd$ is a four-components unconnected graph containing two a components, b component and d component. The letters in the graph string come in alphabetic order.

The F_0 list contains empty graph on 0 nodes, if and only if $|\mathbb{D}| \geq minSup$. The F_1 list contains all one-components frequent graphs or all connected frequent graphs in other words. The F_1 list can be filled by any frequent graphs mining algorithm which is *Gaston*[8] in our case. Let's assume that $F_1 = \{a, b, c, d, e\}$. The two-components candidates come from joining frequent components from F_1 with each other: K_2=$(aa, ab, ac, ad, ae, bb, bc, bd, be, cc, cd, ce, dd, de, ee)$. The support of every candidate C is calculated with subgraphs isomorphism tests over intersection of supporting sets of components which were used to create C, e.g. support of ab is calculated in the $a.suppSet \cap b.suppSet$ set. Please note that unlike sets, graphs do not hold the property that if $i \in D$ supports both a and b than it supports also ab which is presented in fig. 1.

Let's assume that after support calculation of all K_2 candidates we got $F_2 = (aa, ab, ae, bb, bd, cd, ce, de)$. The three-components candidates come from joining all pairs of graphs from F_2 which have the same first components. Therefore: $K_3 = (aaa, aab, aae, abb, abe, bbb, bbd, bdd, cdd, cde, cee, dee)$. The general rule for generating k-components candidates from k-1-components frequent graphs is following.

The K_k list contains graphs which come from joining all pairs of graphs G_i, G_j from F_{k-1}, which contains the same first k-2 components. The result of joining graph G_i=$(C_1, C_2, \ldots C_{k-2}, C_{k-1})$ with graph G_j=$(C_1, C_2, \ldots C_{k-2}, C'_{k-1})$ is graph G=$(C_1, C_2, \ldots C_{k-2}, C_{k-1}, C'_{k-1})$. In example, the result of joining $abbcd$ with $abbcf$ is $abbcdf$.

4 Filtering Based on Connected Frequent Graphs Lattice

There is partially ordering relation of subgraph isomorphism in the set of connected frequent graphs F_1. This relation creates lattice which can be used to filter candidate lists K_i before time consuming support calculation. Filtering is based on the following property:

Property 2. If the process of joining graph G=$(C_1, \ldots C_n)$ with the graph $(C_1, \ldots C_{n-1}, C_{n+1})$ results in infrequent graph, then every joining of the graph G with

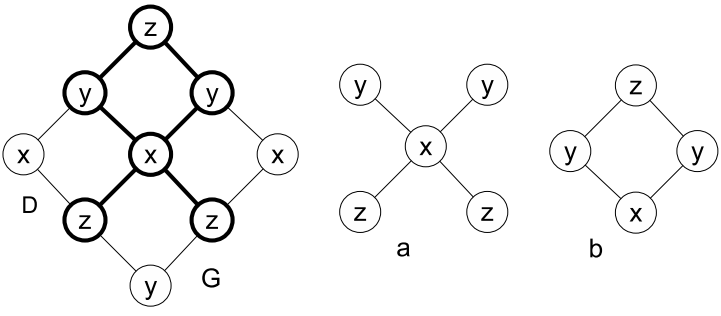


Fig. 1. Graph G supports both a and b, but it does not support ab

the graph $(C_1, \ldots C_{n-1}, C)$ such that C_{n+1} is subgraph isomorphic with C (C is above C_{n+1} in the lattice) results in infrequent graph.

Example 1. Let's assume that in above property $G = abc$. We can now say that if joining of the graph abc with the graph abd results in infrequent $abcd$ graph, then joining of the graph abc with the abx such that d is subgraph isomorphic with x, results in infrequent graph $abcx$.

We use this property to filter candidate list. Let's assume that a given frequent graph G_1 is going to be subsequently joined with frequent graphs $G_1, \ldots, G_n$ (these graphs contains all but last the same components). For this series of joins we prepare the NEG set which will contain last components of graphs $G_1, \ldots, G_n$ such that joining of graph G_1 with these graphs results in infrequent graphs. Before each join of G_1 with G_i, i=2 $\ldots$ n we will check whether NEG contains a graph which is subgraph isomorphic with the last component of G_i. If it contains we do not have to perform join because it would result in infrequent graph. If not - we join G_1 with G_i, calculate support of the result and if it is not frequent we add the last component of G_i to the NEG set.

One can notice that efficiency of this method depends on the joining order. The given graph should be in first place joined with graphs which last component is subgraph isomorphic with last components of graphs joined later. Therefore we propose to order the F_1 list descending on the $f(C)$ key, where $f(C)$ is a number of graphs from F_1 the C is subgraph isomorphic with.

5 The UFC Algorithm

The UFC algorithm (algorithm 1) begins with discovery of all connected frequent graphs by means of any algorithm of this purpose. The discovered graphs are then descendingly ordered by the f key, which was mentioned in the previous section. This creates the F_1 list - the list of frequent one-components graphs. The F_0 list contains the empty graph on 0 nodes iff $|\mathbb{D}| \geq minSup$. Graphs from F_0 and F_1 are in the result list R as well. The F_2 list is created from F_1 list

with *GenerateKComponentsFrequentGraphs* and subsequently - F_{k+1} is created from F_k if F_k is not empty. Graph from all F_i lists are added to the result list R.

Algorithm 1. UFC($\mathbb{D}, minSup$)

if $|\mathbb{D}| \geq minSup$ **then**
  $F_0 \leftarrow$ {null graph on 0 vertices};
$F_1 \leftarrow ConnectedGraphMiner(\mathbb{D}, minSup) -$ {null graph on 0 vertices};
Sort F_1 descendingly on the f key;
$R \leftarrow F_0 \cup F_1$; $i \leftarrow 1$;
while $F_i \neq \emptyset$ **do**
  $i \leftarrow i + 1$; $F_i \leftarrow GenerateKComponentsFrequentGraphs(F_{i-1}, i)$; $R \leftarrow R \cup F_i$;
return R;

The $GenerateKComponentsFrequentGraphs(F_{k-1}, k)$ method returns all k-components frequent graphs, based on F_{k-1} list (k-1 components frequent graphs). The method is consisted of two nested loops: outer loop iterates over graphs G_1 from F_{k-1}, and inner loop iterates over those graphs G_2 from F_{k-1} which can be joined with G_1 - such graphs are always directly next to each other in F_{k-1} list. For each G_1, right before entering the inner loop, we initialize the NEG set. When the joining of G_1 and G_2 results in infrequent graph, the last component of G_2 is added to the NEG set. Before the each joining of G_1 and G_2 we test the condition presented in property 2, that is: if the last component of G_2 contains a subgraph isomorphic to any graph from NEG set, then the joining of G_1 with G_2 is not frequent. The algorithm does not perform the joining of such graphs. Otherwise, the joining of G_1 and G_2 is performed, creating G graph which contains all components of G_1 and last component of G_2. The *isFrequent* method checks if G is frequent, and if it is - it is added to F_k list. Otherwise, the last component of G s added to NEG set.

The *isFrequent* method checks if the G is supported by at least $minSup$ graphs from the $\mathbb{D}$ set which is the intersection of supporting sets of G_1 and G_2. If $|\mathbb{D}| < minSup$ the method returns with false value. Otherwise the support value of G is calculated by executing subgraph isomorphism tests of G with each graph from $\mathbb{D}$ set.

6 Reuse of UGM Optimizations

In [9] we proposed four methods to optimize UGM algorithm. Three of them can be used to optimize UFC algorithm. These are:
- *maximal edge sets check*: The edge set of frequent graph (definition 10) must be a subset of some maximal frequent edgeset. Before support calculation of a candidate we check if its multiset of edge descriptor holds this condition. If not we can be sure that it is infrequent and skip the time consuming support calculation phase.

Algorithm 2. GenerateKComponentsFrequentGraphs(F, k)

```
for i ← 1 to |F| do
  NEG ← ∅;
  innerLoop:
  for j ← i to |F| do
    G1 ← F[i]; G2 ← F[j]; /*G1 = (C1^1, C2^1, ... Ck-1^1), G2 = (C1^2, C2^2, ... Ck-1^2)*/
    if G1 and G2 are identical disregarding last connected-component then
      for all Gneg ∈ NEG do
        if Gneg is subgraph isomorphic to Ck-1^2 then
          continue innerLoop;
      G ← (C1^1, C2^1, ... Ck-1^1, Ck-1^2);
      if isFrequent(G, minSup, G1.supportingSet ∩ G2.supportingSet) then
        R ← R ∪ {G};
      else
        NEG ← {Ck-1^2};
    else
      break;
return R;
```

- *negative border*: The supergraph of infrequent graph is infrequent. Graphs which are found infrequent during support calculation are added to negative border set $\mathbb{GN}$. Before support calculation of a candidate we check if it contains subgraph which is isomorphic to any graph from $\mathbb{GN}$. If so - it is infrequent.
- *break in support calculation*: The exact support value of infrequent graph is unnecessary. If during support calculation of the candidate we can infer that it is not frequent, then we can break this process. The support of a candidate C is calculated with subsequent subgraph isomorphism test over graphs from $\mathbb{D}$ set, which is the intersection of supporting sets of graphs that created C. When the number of negative tests exceeds $|\mathbb{D}|$-$minSup$, then the support of C is less than $minSup$, so we can break the process.

In order to include these optimizations to the UFC algorithm it is sufficient to modify the $isFrequent$ method only. We will call this modified method $isFrequentEx$.

Algorithm 3. isFrequent($G, minSup, \mathbb{D}$)

```
if |D| < minSup then
  return false;
sup ← 0;
for all Gi ∈ D do
  if G is subgraph isomorphic with Gi then
    sup ← sup + 1;
return sup ≥ minSup;
```

Algorithm 4. $isFrequentEx(G, minSup, \mathbb{D})$

```
if |𝔻| < minSup then
    return false;
if ES(G) is not subset of any maximal frequent multisets then
    return false;
if any graph from 𝔾ℕ is subgraph isomorphic with G then
    return false;
sup ← 0; falseResults = 0;
for all G_i ∈ 𝔻 do
    if G is subgraph isomorphic with G_i then
        sup ← sup + 1;
    else
        falseResults ← falseResults + 1;
        if falseResults > |𝔻| − minSup then
            return false;
if sup < minSup then
    𝔾ℕ ← 𝔾ℕ ∪ {G};
    return false;
else
    return true;
```

7 Experiments

Experiments were run on ParMol[1] framework. We used chemical datasets coming from NCI[2], PTC[3] and $MUTAG$[4] collections. We compared the UFC algorithm with two algorithms: UGM and the *gSpanUnconnected* which was briefly proposed in [12].

Table 1 presents execution time for different values of *minSup*. One can notice that: (1) execution time grows extremely fast with the lowering of *minSup*; (2) the UGM is about 50% slower than the UFC and the *gSpanUnconnected* is order of magnitude slower than the UFC; (3) the lower value of *minSup* the bigger is the advantage of UGM and UFC over *gSpanUnconnected*.

Figure 2 presents execution time of five versions of UFC algorithm. The following versions have been tested:

- UFC - the algorithm with all optimizations enabled,
- UFC_0 - the algorithm with all optimizations disabled,
- UFC_negative - the algorithm with the *negative border* optimization enabled,
- UFC_break - the *break in support calculation* optimization enabled,
- UFC_edgeset - the *maximal edge sets check* optimization enabled,

[1] http://www2.informatik.uni-erlangen.de/Forschung/Projekte/ParMol/
[2] http://cactus.nci.nih.gov/ncidb2/download.html
[3] http://www.predictive-toxicology.org/ptc/
[4] ftp://ftp.ics.uci.edu/pub/baldig/learning/mutag/INFO.txt

Table 1. Execution time *UGM*, *UFC* and *gSpanUnconnected* in graph sets from PTC, NCI and MUTAG collections

	execution time [s]													
	ugm					ufc					gspan unconnected			
	minSup [%]					minSup [%]					minSup [%]			
	10	5	4	3	2	10	5	4	3	2	10	5	4	
ptc_FM	5	28	62	510	7158	5	24	43	237	14603	52	366		
ptc_FR	6	29	89	417		5	24	59	200		52	379		
ptc_MM	4	24	77	444	5743	5	19	51	197	4757	46	337		
ptc_MR	6	31	76	726		5	26	52	446		57	414		
	minSup [%]					minSup [%]					minSup [%]			
	30	25	20	15	10	30	25	20	15	10	30	25	20	
nci_786_0	19	35	83	310	4135	19	35	73	250	2679	132	301	768	
nci_A498	19	37	88	340	5080	19	36	76	265	3310	149	333	841	
nci_A549_ATCC	20	36	85	323	4523	19	35	77	259	2992	137	311	806	
nci_ACHN	18	35	82	297		18	35	73	235		139	306	770	
nci_BT_549	16	31	72	297	4741	15	29	61	219	2766	120	274	663	
nci_CAKI_1	18	36	87	343		18	36	77	265		133	319	848	
nci_CCRF_CEM	20	47	89	361	5995	21	34	76	282	3980	131	307	796	
nci_COLO_205	19	38	89	356		19	36	79	281		137	330	843	
nci_DLD_1	12	22	79	961		8	31	52	594		64	181	733	
nci_DMS_114	11	23	87	1097		8	16	55	647		70	189	786	
nci_DMS_273	9	23	99	1346		8	16	62	827		69	208	889	
nci_DU_145	19	34	78	293		18	31	64	217		126	294	715	
nci_EKVX	19	37	88	336		19	36	79	272		139	326	838	
nci_HCC_2998	18	36	95	477	11032	16	34	77	344	6618	128	320	858	
nci_HCT_116	18	35	86	326	4451	18	36	78	259	3014	136	317	807	
nci_HCT_15	19	36	86	325	4331	18	36	78	256	2875	138	323	821	
nci_HL_60_TB	18	34	84	349		16	33	72	258		128	301	790	
nci_HOP_18	9	24	90	1182		7	16	59	689		69	201	855	
nci_HOP_62	18	32	73	274	3588	18	35	74	232	2548	130	302	759	
nci_HOP_92	19	37	90	357	5569	18	36	79	280	3580	137	333	851	
nci_HS_578T	16	31	69	256	3248	16	31	64	203	2097	126	292	700	
nci_HT29	19	36	89	334		19	37	79	266		141	323	841	
nci_IGROV1	19	37	87	337		18	36	78	265		139	325	811	
nci_KM12	21	40	87	339	5090	19	37	78	266	3203	140	336	830	
nci_KM20L2	10	24	96	1298		8	17	61	798		72	208	873	
nci_K_562	18	35	82	301	4152	18	35	74	236	2645	133	310	768	
nci_LOX_IMVI	19	36	84	326	4871	18	35	74	252	3197	137	312	808	
nci_LXFL_529	8	17	58			6	12	35			57	158		
nci_M14	19	35	83	305	3815	18	36	74	235	2354	138	319	1350	
nci_M19_MEL	10	24	92	1126		8	17	59	694		71	204	1666	
nci_MALME_3M	17	33	78	295	4350	16	32	71	230	2844	121	274	1308	
	minSup [%]					minSup [%]					minSup [%]			
	50	40	30			50	40	30			50	40	30	
mutag_188	16	21	481			19	27	465			242	335		

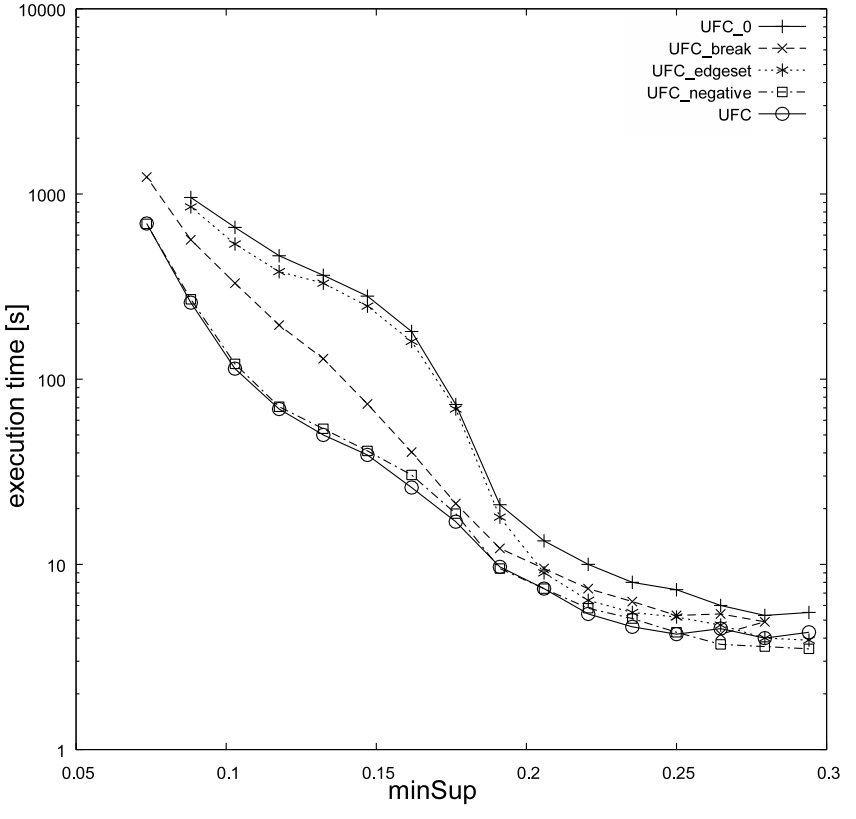


Fig. 2. Execution time of UFC with and without optimizations run on Chemical_340 dataset form NCI collection

The figure 2 shows that: (1) all optimization give significant boost to the UFC algorithm; (2) the UFC algorithm is about ten times faster than the UFC_0; (3) the *negative border* optimization is the most efficient.

8 Conclusions

We have proposed a new algorithm called UFC for finding both connected and unconnected frequent graphs. It makes use of frequent connected graphs which are discovered by any algorithm for this purpose and then joins these connected graphs in *Apriori*-like method to create frequent unconnected graphs. Such approach was said to be inefficient in [9] but with the aid of UGM optimizations and proposed candidate filtering we made it faster than the UGM and $gSpanUnconnected$ algorithms.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD 1993, pp. 207–216 (1993)
2. Borgelt, C., Berthold, M.R.: Mining Molecular Fragments: Finding Relevant Substructures of Molecules. In: ICDM, vol. 00:51 (2002)
3. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraph in the presence of isomophism (2003)
4. Huan, J., Wang, W., Prins, J., Yang, J.: SPIN: mining maximal frequent subgraphs from graph databases. In: KDD 2004, pp. 581–586 (2004)
5. Inokuchi, A., Washio, T., Motoda, H.: An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data (2000)
6. Inokuchi, A., Washio, T., Nishimura, K., Motoda, H.: A fast algorithm for mining frequent connected subgraphs. Research Report RT0448 (2002)
7. Kuramochi, G., Karypis, M.: Frequent subgraph discovery. In: Proceedings IEEE International Conference on Data Mining, 2001, ICDM 2001, pp. 313–320 (2001)
8. Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. In: KDD 2004, pp. 647–652 (2004)
9. Skonieczny, L.: Mining of Unconnected Frequent Graphs with Direct Subgraph Isomorphism Tests. In: ICMMI 2009 (2009)
10. Ting, R., Bailey, J.: Mining minimal contrast subgraph patterns. In: Jonker, W., Petković, M. (eds.) SDM 2006. LNCS, vol. 4165, pp. 639–643. Springer, Heidelberg (2006)
11. Yan, X., Han, J.: gSpan: Graph-based substructure pattern mining. In: ICMD, vol. 00:721 (2002)
12. Yan, X., Han, J.: CloseGraph: mining closed frequent graph patterns. In: KDD 2003, pp. 286–295. ACM , New York (2003)

Distributed Classification for Pocket Data Mining

Frederic Stahl[1], Mohamed Medhat Gaber[1], Han Liu[1], Max Bramer[1], and Philip S. Yu[2]

[1] School of Computing, University of Portsmouth
Portsmouth, PO1 3HE, UK
[2] Department of Computer Science, University of Illinois at Chicago
851 South Morgan Street, Chicago, IL 60607-7053, USA

Abstract. Distributed and collaborative data stream mining in a mobile computing environment is referred to as Pocket Data Mining *PDM*. Large amounts of available data streams to which smart phones can subscribe to or sense, coupled with the increasing computational power of handheld devices motivates the development of *PDM* as a decision making system. This emerging area of study has shown to be feasible in an earlier study using technological enablers of mobile software agents and stream mining techniques [1]. A typical *PDM* process would start by having mobile agents roam the network to discover relevant data streams and resources. Then other (mobile) agents encapsulating stream mining techniques visit the relevant nodes in the network in order to build evolving data mining models. Finally, a third type of mobile agents roam the network consulting the mining agents for a final collaborative decision, when required by one or more users. In this paper, we propose the use of distributed Hoeffding trees and Naive Bayes classifiers in the *PDM* framework over vertically partitioned data streams. Mobile policing, health monitoring and stock market analysis are among the possible applications of *PDM*. An extensive experimental study is reported showing the effectiveness of the collaborative data mining with the two classifiers.

1 Introduction

Recent and continuous advances in smart mobile devices have opened the door for running applications that were difficult or impossible to run in the past in such resource-constrained environments. The clear trend is to have more applications running locally on these devices given their computational and sensing capabilities. Recent important applications in the area of activity recognition [9,14] have stimulated our recent research activities. Therefore, we have proposed the new area of pocket data mining in [1].

Pocket data mining has been first coined in [1] to describe the process of mining data streams collaboratively in a mobile computing environment. The *PDM* framework supports the process from resource discovery to the learning and usage phases. The core of the framework is the set of data stream mining

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 336–345, 2011.

techniques that work together to synthesize a global outcome. The mining techniques are assigned to the mobile devices and are encapsulated as mobile software agents that are able to move and be cloned. This assignment is done based on the availability of resources and features of the data available to the mobile device. We have proved in [1] the computational feasibility of the *PDM* framework. However, we have not employed any stream mining technique in our previous implementation. Thus, in this paper, we propose the use of two data stream classification techniques. We have chosen Hoeffding trees and Naive Bayes classifiers due to the following reasons. First, Hoeffding trees classifiers have proved their efficiency as the state-of-the-art data stream classification technique as reported in [3]. Second, Naive Bayes is a lightweight naturally incremental classifier.

The paper is organised as follows. Section 2 enumerates the related work. The architecture of the PDM framework including the details of the used algorithms is given in Section 3. Extensive experimental study is presented in Section 4. Ongoing work and concluding remarks and be found in Section 5.

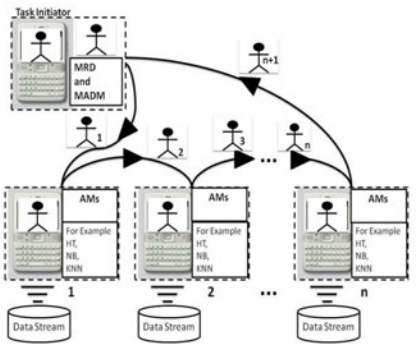
2 Related Work

Distributed data mining techniques have been thoroughly reviewed by Park and Kargupta in [16]. On the other hand, the field of data stream mining has been concisely reviewed in [11]. More recently, the utilisation of smart phones' sensing capabilities to be used to learn about the user's activities have been explored in [13,9]. It has to be noted that none of the above techniques has explored the great potential of collaborative mining of data streams in the mobile ad hoc computing environments, including the work proposed by Miller et el [15], that only focused on recommender systems for any type of connected devices. Our work rather attempts to exploit data stream mining techniques for ad hoc analysis using smartphones in critical applications.

3 PDM: The Pocket Data Mining Architecture

The architecture of the *PDM* framework is highlighted in Figure 1. The basic scenario of *PDM* displayed in Figure 1 involves the following generic (mobile) software agents [1]: (a) (Mobile) **A**gent **M**iners (AM) are distributed over the network and located on the local mobile devices, they implement data mining algorithms; (b) **M**obile Agent **R**essource **D**iscoverers are used to explore the network and locate computational ressources, data sources and AMs; (c) **M**obile **A**gent **D**ecision **M**akers roam the network consulting AMs in order to retrieve information or partial results for the data mining task. It has to be noted that we use the terms *PDM architecture* and *PDM framework* interchangeably. Any smart phone in the network can own any kind of *PDM* agents. The smart phone from which a data mining task is initiated is called the task initiator. The AMs in *PDM* can implement any data mining algorithm such as Hoeffding Decision Trees (HT), Naive Bayes (NB) or K Nearest Neighbours (K-NN). As the left hand side of Figure 1 shows, these data mining algorithms are embedded in AMs which

run on-board a users smart phone. The smart phone may be subscribed to a data stream. The data mining model that is generated by the AMs is continuously updated in order to cope with possible concept drifts of the data stream. AMs may be stationary agents already installed for the use by the owner of the smart phone but may also be mobile agents distributed at the beginning of the data mining task. The right hand side of Figure 1 shows the pseudo code of the basic PDM workflow.



Algorithm 1 PDM's collaborative data mining workflow

Task Initiation: Form an ad hoc network of mobile phones;
Task Initiation: start *MRD* agent;
MRD: Discover data sources, computational resources and techniques;
MRD: Decide on the best combination of techniques to perform the task;
MRD: Decide on the choice of stationary AMs and deploy mobile AMs;
Task Initiation: start *MADM* agent with schedule provided by the *MRD*;
for $i = 1$ **to** $i =$ *number of AMs* **do**
  repeat
    AM_i: mine streaming data;
  until Use of the model by *MADM*
end for

Fig. 1. PDM Architecture

For example in the context of classification, if the task initiator at any point in time decides to use the models of remotely located AMs to classify a set of unlabelled data instances, an MADM and an MRD can be used. The MRD is roaming the network to discover available AMs and data streams onboard the smart phones. The MADM then loads the unlabelled instances and visits relevant AMs, as determined by the MRD, in order to collect their predictions for the correct class labels. While the MADM agent visits different nodes in the network, it might decide to terminate its itinerary based on a stopping criterion such as confidence level of the already collected predictions, or a time limit.

The current implementation, which is evaluated in the paper, has two different AMs for classification tasks, namely Hoeffding Tree [3] and Naive Bayes classifiers. However, the *PDM* framework allows the use of any classification technique. The Hoeffding tree classifier from the *MOA* tool as illustrated by Bifet and Kirkby in [2] is shown in Figure 2. Hoeffding tree classifiers have been designed for high speed data streams. On the other hand, the Naive Bayes classifier has been developed for batch learning, however it is naturally incremental. The current implementation of *PDM* uses the Naive Bayes classifier from the *MOA* tool [2] which is based on the Bayes Theorem [14] stating that if $P(C)$ is the probability that event C occurs and $P(C|X)$ is the conditional probability that event C occurs under the premise that X occurs then $P(C|X) = \frac{P(X|C)P(C)}{P(X)}$. According to the Bayes Theorem, the Naive Bayes algorithm assigns a data instance to the class it belongs to with the highest probability. As Naive Bayes

Algorithm 2 Hoeffding tree induction algorithm.

1: Let HT be a tree with a single leaf (the root)
2: **for all** training examples **do**
3: Sort example into leaf l using HT
4: Update sufficient statistics in l
5: Increment n_l, the number of examples seen at l
6: **if** $n_l \bmod n_{min} = 0$ **and** examples seen at l not all of same class **then**
7: Compute $\overline{G}_l(X_i)$ for each attribute
8: Let X_a be attribute with highest $\overline{G}_l$
9: Let X_b be attribute with second-highest $\overline{G}_l$
10: Compute Hoeffding bound $\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n_l}}$
11: **if** $X_a \neq X_\emptyset$ **and** $(\overline{G}_l(X_a) - \overline{G}_l(X_b) > \epsilon$ **or** $\epsilon < \tau)$ **then**
12: Replace l with an internal node that splits on X_a
13: **for all** branches of the split **do**
14: Add a new leaf with initialized sufficient statistics
15: **end for**
16: **end if**
17: **end if**
18: **end for**

Fig. 2. Hoeffding Tree Algorithm

generally performs well [10,12] and is naturally incremental, it is suitable for classifying data streams.

4 Implementation and Evaluation

As this paper examines PDM's applicability to classification rule induction on data streams, both Hoeffding tree and Naive Bayes classifiers have been thoroughly tested.

4.1 Experimental Setup

We use the implementation of both classifiers in *MOA* toolkit [2] which is based on the WEKA library [4]. We compare two *PDM* configurations, one where all AMs are based on Hoeffding trees and the other where all AMs are based on Naive Bayes. Owners of different AMs may have subscribed to overlapping subsets of the feature space of the same data stream, as there may be features that are particularly interesting for the owner of a local AM. However, the current subscription may be insufficient for classifying new data instances. Subscribing to more features may not be desirable for many reasons, such as that it may lead to higher subscription fees or confidentiality constraints. However the owner of the local AM sends an MADM that visits and consults further AMs that belong to different owners. The visited AMs are potentially subscribed to different features and the MADM consults the AMs to classify the unlabelled data instances. The classification results and accompanying information from the local AMs are collected and used by the MADM to decide for a final classification. The accompanying information of the AMs is an estimate of the AM's own confidence which is referred to as 'weight' in this paper. In *PDM*, each AM takes with a previously defined probability a labelled streamed data instance as training or as test instance. In the current setup the probability that an instance is selected

as test instance is 20% and as training instance is 80%. The test instances are used by the local AM to estimate its local classification accuracy/confidence or 'weight'. Concept drifts are also taken into account when the weight is calculated. This is done by defining a maximum number of test instances at the startup of an AM by the owner. For example, if the maximum number of test instances is 20, and already 20 test instances have been selected then the oldest test instance is replaced by the next newly selected test instance and the 'weight' is recalculated.

The MADM hops with the instances to be classified to each available AM, requesting to classify the instances and also retrieves the AM's weight. After the MADM's schedule is processed, the MADM derives the final classification for each data instance by 'weighted majority voting'. For example, if there are three AMs A, B and C and one data instance to classify, AM A predicts class X with a 'weight' of 0.55, AM B predicts class X with a 'weight' of 0.2 and AM C predicts class Y with a 'weight' of 0.8, then MADM's 'weighted' prediction would be for class X $0.55 + 0.2 = 0.75$ and for class Y 0.8 and as Y yielded the highest vote, the MADM would label the instance with class Y.

For the implementation of *PDM* the well known JADE framework has been used [5], with the reasoning that there exist a version of JADE, JADE-LEAP (Java Agent Development Environment-Lightweight Extensible Agent Platform), that is designed for the implementation of agents on mobile devices and can be retrieved from the JADE project website as an 'add on' [6]. As JADE works on standard PCs as well as on mobile devices, it is possible to develop and evaluate the *PDM* framework on a test LAN. The LAN consists of 8 PCs with different hardware configurations, which are connected using a standard CISCO System switch of the catalyst 2950 series. In our experimental setup, we used 8 AMs each running a Hoeffding tree induction algorithm or Naive Bayes, and one MADM collecting classification results. The AMs in the current implementation can be configured so that they only take specific features into account or a particular percentage of randomly selected features out of the total number of features. The latter configuration is for our experimental purposes as in the real application we may not know which features a certain AM is subscribed to.

Table 1. Evaluation Datasets

Test Number	Dataset	Number of Attributes
1	kn-vs-kr	36
2	spambase	57
3	waveform-500	40
4	mushroom	22
5	infobiotics 1	20
6	infobiotics 2	30

The data streams were simulated using the datasets described in Table 1. Datasets for tests, 1, 2, 3 and 4 were retrieved from the UCI data repository [7] and datasets for tests 5 and 6 were retrieved from the Infobiotics benchmark

data repository [8]. The simulated data stream takes a random data instance from the dataset and feeds it to the AM. Instances might be selected more than once by the data stream, however if a data instance has been used as a test instance it will be removed from the stream and never selected again, in order to avoid overfitting of the AM's model and calculate the 'weight' on test instances.

4.2 Distributed Hoeffding Trees

The fact that the datasets in Table 1 are batch files allows us to compare PDM's accuracy with Hoeffding trees to batch learning classification algorithms, in particular PDM's accuracy is compared to accuracy of C4.5 and the accuracy of Naive Bayes. Both batch implementations were retrieved from WEKA [4]. The choice of C4.5 is based on its wide acceptance and use; and to the fact that the Hoeffding tree algorithm is based on C4.5. The choice of Naive Bayes is based on the fact that it is naturally incremental, computationally efficient and also widely accepted.

In general, the more features an AM has available and the more AMs visited, the more likely it is to have a better accuracy of the global classification. However, some features may not be relevant to the classification task and introduce unnecessary noise. For all experiments in this section, 30% of the data is taken as test instances for the MADM and the remaining 70% for training the AMs. All experiments have been conducted 5 times and the achieved local accuracies on the AMs and the achieved accuracy of the MADM has been recorded and averaged.

The left hand side of Figure 3 shows the accuracy of *PDM* plotted versus the number of AMs visited by the MADM. The experiments have been conducted for AM's holding a percentage of features from the total feature space, in particular 20%, 30% and 40% of the total feature space. The features an AM holds have been randomly selected for these experiments, however it is possible that different AMs may have selected the same or partially the same features. Looking at the left hand side of Figure 3, it can be seen that the batch versions of C4.5 and Naive Bayes achieve a comparable accuracy on all datasets. The largest discrepancy between both algorithms is for Test 2 where Naive Bayes's accuracy was 80% and C4.5 91%. Regarding PDM's classification accuracy, it can be seen that in all cases the achieved accuracy is no less than 50%. In general, it can be observed that for configurations of *PDM* that use AMs with only 20% of the attributes, PDM's classification accuracy is low compared with configurations that use 30% or 40% of the attributes. Also there does not seem to be a large discrepancy between configurations that use 30% or 40% of the attributes, which may be due to the fact that it is more likely with 40% attributes that irrelevant attributes have already been selected. In general, it can be seen that in many cases PDM's classification accuracy is close to the batch classification accuracy of C4.5 and Naive Bayes, especially for tests 3 and 5, however also for the remaining tests *PDM* often achieves close accuracies compared to those achieved from the batch learning algorithms. In general PDM with Hoeffding Trees achieves an acceptable classification accuracy. The right hand side of Figure 3 shows the

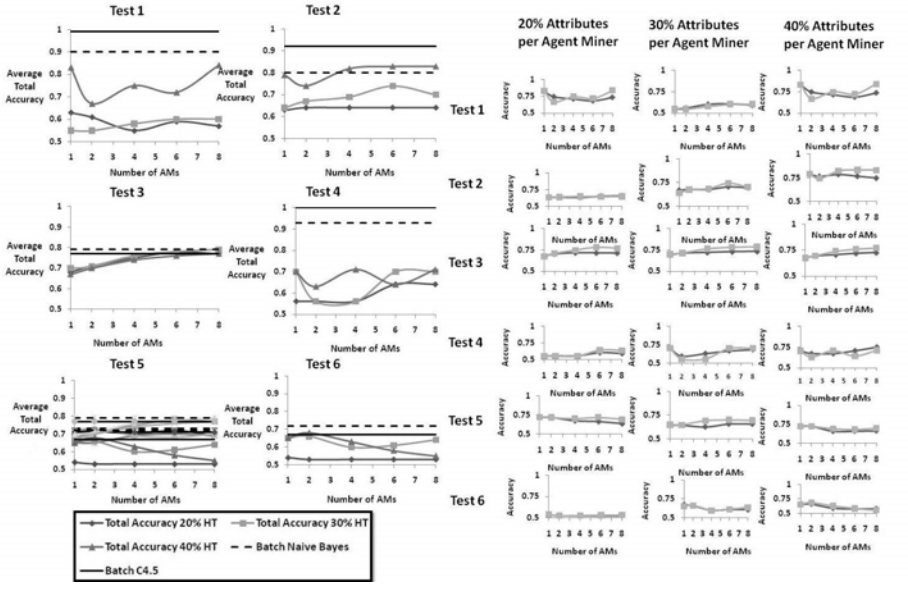


Fig. 3. The left hand side of the figure shows PDM's classification accuracy based on Hoeffding Trees and the right hand side of the figure the average accuracy of the MADM (using 'weighted' majority voting) versus the average accuracy of the AMs based on PDM with Hoeffding Trees

accuracy achieved by the MADM (using 'weighted' majority voting) and the average of the local accuracies achieved by the AMs versus the number of AMs that have been visited. Each row of plots on the right hand side in Figure 3 corresponds to one of the datastets listed in Table 1 and each column of plots corresponds to a different percentage of features subscribed to by the AMs. The darker lines in the plots correspond to the average accuracy of the AMs and the lighter lines correspond to the accuracy the MADM derived using the local AM's 'weights' and classifications. It can be observed that in most cases the 'weighted' majority voting either achieves a similar or better accuracy compared with simply taking the average of the predictions from all AMs.

4.3 Distributed Naive Bayes

Similarly, *PDM* using Naive Bayes has been evaluated the same way as Hoeffding Trees described in Section 4.2. Similar results compared with Section 4.2 are expected with *PDM* using Naive Bayes classifiers.

Figure 4 illustrates the data obtained with *PDM* using Naive Bayes the same way as in Figure 3. The left hand side of Figure 4 shows the total accuracy of *PDM* plotted versus the number of AMs visited by the MADM. Regarding *PDM*'s classification accuracy, again it can be seen that in all cases the achieved accuracy is not less than 50%. In general, it can be observed that for configurations of PDM

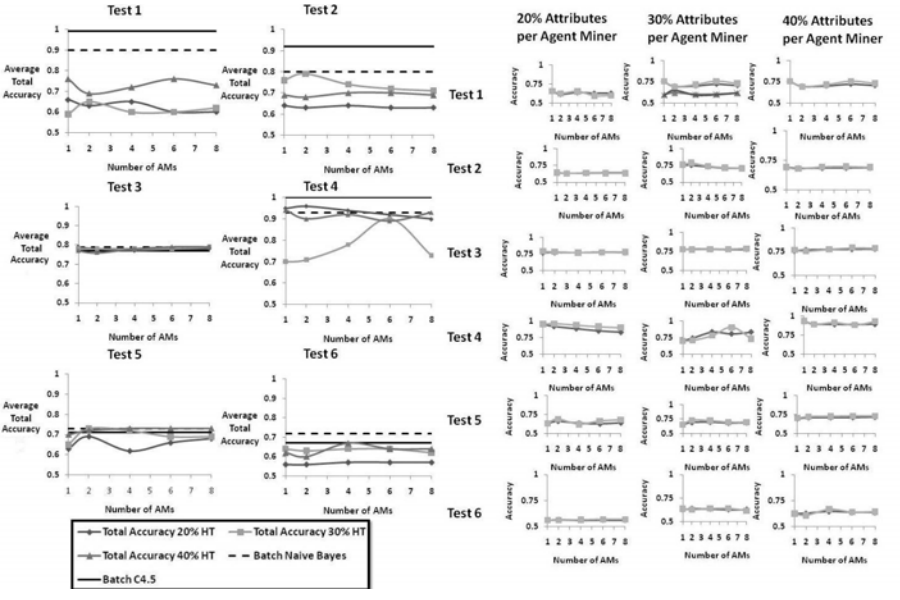


Fig. 4. The left hand side of the figure shows PDM's classification accuracy based on Naive Bayes and the right hand side of the figure shows the average accuracy of the MADM (using 'weighted' majority voting) versus the average accuracy of the AMs based on PDM with Naive Bayes

that use AMs with only 20% of the attributes PDM's classification accuracy is low compared with configurations that use 30% or 40% of the attributes. In general, it can be seen that in most cases that PDM's classification accuracy is close to the batch classification accuracy of C4.5 and Naive Bayes, especially for tests 3, 4, 5 and 6, however also for the remaining tests *PDM* often achieves acceptable accuracies. The achieved accuracies are close compared with those achieved by the batch learning algorithms which have the advantage over *PDM* of having all the features available. In general *PDM* with Naive Bayes AMs achieves an acceptable classification accuracy. The right hand side of Figure 4 shows the accuracy achieved by the MADM (using 'weighted' majority voting) and the average of the local accuracies achieved by the AMs versus the number of AMs that have been visited. Each row of plots in Figure 4 corresponds to one of the datasets listed in Table 1 and each column of plots corresponds to a different percentage of features subscribed to by each AM. The darker lines in the plots correspond to the average accuracy of the AMs and the lighter lines correspond to the accuracy the MADM derived using the local AM's 'weights' and classifications. Similar to the Hoeffding tree results, It can be observed that in most cases the 'weighted' majority voting either achieves a similar or better accuracy compared with simply taking the average of the predictions from all AMs.

The bar charts in Figure 5 show for each number of used AMs the accuracies of *PDM* in the following order from left to right: Total accuracy of PDM with Hoeffding Trees with 20% attributes; total accuracy of *PDM* with Hoeffding Trees with

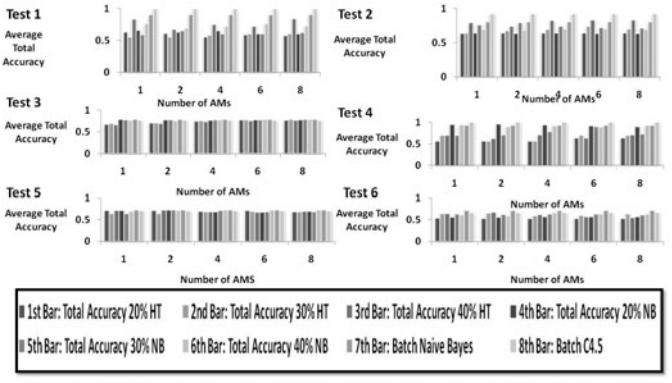


Fig. 5. Classification Accuracies achieved by both, PDM with Hoeffding Trees and PDM with Naive Bayes

30% attributes; total accuracy of PDM with Hoeffding Trees with 40% attributes; total accuracy of PDM with Naive Bayes with 20% attributes; total accuracy of PDM with Naive Bayes with 30% attributes; total accuracy of PDM with Naive Bayes with 40% attributes; accuracy for batch learning of Naive Bayes with all attributes; and finally accuracy for batch learning of C4.5 with all attributes. For tests 3 and 5, PDM with Naive Bayes AMs and PDM with Hoeffding tree AMs seem to achieve an equal performance concerning the classification accuracy. In the remaining tests 1, 2, 4 and 6, there seems to be no general bias towards one of the two approaches, sometimes PDM with Hoeffding tree AMs is slightly better than PDM with Naive Bayes AMs and vice versa. The fact that there doesn't seem to be a bias towards one of the approaches suggest that heterogeneous PDM configurations with some AMs implementing Naive Bayes and some implementing Hoeffding trees would generate a similar performance compared with PDM systems solely based on Naive Bayes or Hoeffding trees.

5 Conclusions

This paper outlines the Pocked Data Mining (PDM) architecture, a framework for collaborative data mining on data streams in a mobile environment. *PDM* uses mobile agents technology in order to facilitate mining of data streams collaboratively. *PDM* has been evaluated concerning its achieved classification accuracy for two different configurations, one with Hoeffding Tree AMs and one with Naive Bayes AMs. It has been observed that both configurations achieve an acceptable classification accuracy. Often PDM even achieves close accuracies compared to the ideal case, where all instances and attributes are available in a batch file, so that a batch learning algorithm such as C4.5 or Naive Bayes can be applied. Also it does not seem that in *PDM*, one of the used classifiers (Hoeffding Tree or Naive Bayes) is superior to the other, both setups achieve very similar results. Also it has been observed that PDM's weighted majority

voting achieves a better classification accuracy compared with simply the taking the local average accuracies of all AMs.

PDM opens a powerful yet so far widely unexplored distributed data mining niche. The particular *PDM* implementation outlined in this paper for classification just scratches the surface of possibilities of collaborative data mining on mobile devices.

References

1. Stahl, F., Gaber, M.M., Bramer, M., Yu, P.S.: Pocket Data Mining: Towards Collaborative Data Mining in Mobile Computing Environments. In: Proceedings of the IEEE 22nd International Conference on Tools with Artificial Intelligence (ICTAI 2010), Arras, France, October 27-29 (2010)
2. Bifet, A., Kirkby, R.: Data Stream Mining: A Practical Approach. Center for Open Source Innovation (August 2009)
3. Domingos, P., Hulten, G.: Mining high-speed data streams. In: International Conference on Knowledge Discovery and Data Mining, pp. 71–80 (2000)
4. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 2nd edn. Morgan Kaufmann, San Francisco (2005)
5. Bellifemine, F., Poggi, A., Rimassa, G.: Developing multi-agent systems with JADE. In: Castelfranchi, C., Lespérance, Y. (eds.) ATAL 2000. LNCS (LNAI), vol. 1986, pp. 89–103. Springer, Heidelberg (2001)
6. JADE-LEAP, http://jade.tilab.com/
7. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases (Technical Report). University of California, Irvine, Department of Information and Computer Sciences (1998)
8. Bacardit, J., Krasnogor, N.: The Infobiotics PSP benchmarks repository (April 2008), http://www.infobiotic.net/PSPbenchmarks
9. Choudhury, T., Borriello, G., et al.: The Mobile Sensing Platform: An Embedded System for Activity Recognition. Appears in the IEEE Pervasive Magazine - Special Issue on Activity-Based Computing (April 2008)
10. Domingos, P., Pazzani, M.J.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning 29(2/3), 103–130 (1997)
11. Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining Data Streams: A Review. ACM SIGMOD Record 34(1), 18–26 (2005) ISSN: 0163-5808
12. Hilden, J.: Statistical diagnosis based on conditional independence does not require it. Computers in Biology and Medicine 14(4), 429–435 (1984)
13. Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.: A Survey of Mobile Phone Sensing. IEEE Communications (September 2010)
14. Langley, P., Iba, W., Thompson, K.: An analysis of bayesian classifiers. In: National Conference on Artificial Intelligence, pp. 223–228 (1992)
15. Miller, B.N., Konstan, J., Riedl, J.: PocketLens: Toward a personal recommender system. ACM Transactions on Information Systems 22(3) (2004)
16. Park, B., Kargupta, H.: Distributed Data Mining: Algorithms, Systems, and Applications. In: Ye.N.(ed.) Data Mining Handbook (2002)

K-Means Based Approaches to Clustering Nodes in Annotated Graphs

Tijn Witsenburg[1] and Hendrik Blockeel[1,2]

[1] Leiden Institute of Advanced Computer Science, Universiteit Leiden
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
tijn@liacs.nl, blockeel@liacs.nl
[2] Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, 3001 Leuven, Belgium
hendrik.blockeel@cs.kuleuven.be

Abstract. The goal of clustering is to form groups of similar elements. Quality criteria for clusterings, as well as the notion of similarity, depend strongly on the application domain, which explains the existence of many different clustering algorithms and similarity measures. In this paper we focus on the problem of clustering annotated nodes in a graph, when the similarity between nodes depends on both their annotations and their context in the graph ("hybrid" similarity), using k-means-like clustering algorithms. We show that, for the similarity measure we focus on, k-means itself cannot trivially be applied. We propose three alternatives, and evaluate them empirically on the Cora dataset. We find that using these alternative clustering algorithms with the hybrid similarity can be advantageous over using standard k-means with a purely annotation-based similarity.

Keywords: clustering, graph mining, similarity measure, k-means.

1 Introduction

1.1 Clustering

Clustering is a common data mining task. It can be defined as: given a set of data elements, partition it into subsets ("clusters") such that elements within a subset are highly similar to each other, and elements in different subsets are highly dissimilar. This is a very broad definition: first, one needs to specify the notion of similarity; second, even if this is clearly defined, there is the question of exactly how to measure the quality of the clustering. As a result, many different clustering methods have been proposed, each with their own strengths and weaknesses.

Within clustering, we can distinguish the clustering of elements of a set (where each element has its own independent description, and "similarity" refers to the similarity of these descriptions) and clustering of nodes in a graph (where "similarity" refers to their topological closeness or connectedness in the graph). We call the first setting **standard clustering**, the second **graph clustering**.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 346–357, 2011.

(We use the latter term to be consistent with some of the literature, even if it could be misunderstood as clustering graphs rather than nodes in a graph - we do mean the latter.)

A setting in between these two is where each node in a graph has its own description (here called *annotation*), in addition to being connected to other nodes. We call this setting *annotated graph clustering*. While there has been much research on standard and graph clustering, this mixed setting has only recently started to receive attention, despite its obvious application potential in web mining, systems biology, etc. It is clear that neither standard clustering, nor graph clustering, is optimal in this setting, as each exploits only one part of the available information. We will call any method that exploits both the information in the annotations and in the graph structure a **hybrid clustering** method. It has been shown before that hybrid methods can yield better clustering results [15,16].

In this paper we build on earlier work by Witsenburg and Blockeel [15], who proposed a hybrid similarity measure and showed that it can be used successfully for agglomerative clustering. We investigate to what extent the same similarity measure can be used in k-means-like clustering approaches. It turns out that k-means cannot be used *as is* with this measure, because no suitable center measure can be defined. We propose three alternatives: one is the use of k-medoids instead of k-means, the other two are new variants of k-means. An empirical evaluation shows that these variants can yield a better clustering than plain k-means with a standard similarity measure.

In the following section, we provide some more background and discuss related work. In Section 3, we discuss the hybrid similarity measure we will use, and the k-means algorithm. In Section 4, we discuss the problem with using the hybrid similarity with k-means, which boils down to the lack of a good center measure, and we present three ways in which this problem can be solved. In Section 5 we present experiments showing that the proposed algorithms, with the hybrid measure, can indeed yield better clusters. We conclude in Section 6.

2 Related Work

Standard clustering methods include bottom-up hierarchical clustering methods, the well-known k-means algorithm [9], and many more methods and variants; for an overview, see, e.g., Hartigan [5]. In graph clustering, too, a variety of methods exist (e.g., [1,14,3]); many of these handle weighted edges and construct minimal cuts [2], i.e., the quality criterion for a clustering is that the total weight of connections between clusters should be minimal.

Hybrid methods have not received much attention. One thread of approaches can be found in inductive logic programming, where methods for relational clustering have been proposed [13]. These methods usually consider objects that are assumed to be independent but have an internal structure that is relational. They are typically not set in the context of clustering nodes in a graph. Neville et al. [12] were among the first to explicitly discuss the need for incorporating

node content information into graph clustering. More specifically, they consider graph clustering algorithms that can work with weighted edges, and define the weights according to the similarity of the nodes connected by the edge. Thus, they map the hybrid clustering problem to a graph clustering problem, after which any graph clustering method can be used.

More recently, Zhou et al. [16] proposed a method that inserts nodes in the graph for every attribute value in the annotations. Then, edges connect the original nodes with these *attribute nodes* when this attribute value is in the annotation of this original node. This approach is somewhat more flexible with respect to trading off the different goals of standard clustering and graph clustering similarity; for instance, two nodes that originally did not have a path between them could still be in the same cluster, since they can be connected through one or more attribute nodes. This is not the case for most graph clustering methods.

While the above mentioned approaches reduce the hybrid clustering problem to a graph clustering problem, Witsenburg and Blockeel [15] did the opposite: they incorporated graph information into a standard clustering approach. They proposed a similarity measure that combines similarity of annotations with context information from the graph, and showed that bottom-up hierarchical clustering methods can be made to produce better results by using this new similarity measure, instead of a pure annotation-based one. One advantage of translating to standard clustering is that a plethora of clustering methods become available, more than for the translation to graph clustering.

However, not all of those may be as readily available as it may seem. For instance, k-means clustering does not only require a similarity measure, but also a center measure (sometimes referred to as centroid or prototype definition) that is compatible with it. For Witsenburg and Blockeel's hybrid similarity measure, it is not obvious what that center measure should be. Hence, the hybrid similarity cannot simply be plugged in into k-means. In this paper we discuss and compare three ways around this problem.

3 Background Definitions

We here recall Witsenburg and Blockeel's similarity measures [15] and the k-means algorithm [9].

Content-based, Contextual and Combined Similarity. Consider the data set that needs to be clustered to be an annotated graph, then this data set D can be defined as $D = (V, E, \lambda)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of n vertices or elements, $E \subseteq V \times V$ is the set of edges, and $\lambda : V \to \mathcal{A}$ a function that assigns to any element v of V an "annotation"; this annotation $\lambda(v)$ is considered to be the *content* of vertex v. The graph is undirected and an edge cannot loop back to the same vertex, so with two elements $v, w \in V$ this means that $(v, w) \in E \Leftrightarrow (w, v) \in E$ and $(v, v) \notin E$.

The space of possible annotations is left open; it can be a set of symbols from, or strings over, a finite alphabet; the set of reals; an n-dimensional Euclidean

space; a powerset of one of the sets just mentioned; etc. The only constraint on $\mathcal{A}$ is that it must be possible to define a similarity measure $\mathcal{S}_{content} : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ as a function that assigns a value to any pair of annotations expressing the similarity between these annotations. Since this similarity is entirely based on the content of the vertices, it will be called the *content-based similarity*. Normally the value of this similarity is in the range $[0, 1]$ where 0 stands for no similarity at all and 1 means that they are considered to be identical.

Now let $\phi : V \times V \to \{0, 1\}$ be a function that assigns a value to a pair of elements in the data set such that $\phi(v, w) = 1$ if $(v, w) \in E$ and 0 otherwise. We define the *neighbor similarity* $\mathcal{S}_{neighbor} : V \times V \to \mathbb{R}$ between two elements v and w from V as the average content-based similarity between v and all neighbors of w:

$$\mathcal{S}_{neighbor}(v, w) = \frac{\sum_{u \in V} \mathcal{S}_{content}(\lambda(v), \lambda(u)) \cdot \phi(u, w)}{\sum_{u \in V} \phi(u, w)} \tag{1}$$

This similarity is not symmetric, but we can easily symmetrize it, leading to the *contextual similarity* $\mathcal{S}_{context} : V \times V \to \mathbb{R}$:

$$\mathcal{S}_{context}(v, w) = \frac{\mathcal{S}_{neighbor}(v, w) + \mathcal{S}_{neighbor}(w, v)}{2} \tag{2}$$

The motivation behind defining this similarity measure is that, if similar nodes tend to be linked together, then the neighbors of w in general are similar to w. Hence, if similarity is transitive, a high similarity between v and many neighbors of w increases the reasons to believe that v is similar to w, even if there is little evidence of such similarity when comparing v and w directly (for instance, due to noise or missing information in the annotation of w).

This contextual similarity measure is complementary to the content-based one; it does not use the content-based similarity between v and w at all. Since in practical settings it may be good not to ignore this similarity entirely, Witsenburg and Blockeel also introduced the *combined similarity* $\mathcal{S}_{combined} : V \times V \to \mathbb{R}$:

$$\mathcal{S}_{combined}(v, w) = c \cdot \mathcal{S}_{content}(v, w) + (1 - c) \cdot \mathcal{S}_{context}(v, w) \tag{3}$$

with $0 \leq c \leq 1$. c determines the weight of the content-based similarity in the combined similarity. As Witsenburg and Blockeel [15] found no strong effect of using different values for c, from now on we consider only the combined similarity with $c = \frac{1}{2}$.

Both the contextual and the combined similarity measures are hybrid similarity measures, since they use both content-based and graph information. The contextual similarity measure between two nodes v and w does take the contents of v and w into account, it just does not use the direct similarity between these contents.

Note that any standard clustering method that can cluster nodes based on (only) their content similarity, can also cluster nodes based on the contextual or combined similarity, and in the latter case it implicitly takes the graph structure into account; the method itself need not know that these data come from a graph.

The *k*-means algorithm. k-means [9] is a clustering algorithm that works as follows. Data set elements are assumed to be vectors in an n-dimensional space, and similarity is expressed by Euclidean distance (the smaller the distance, the greater the similarity). The number of clusters k is a parameter of the algorithm. k-means proceeds as follows:

1. Choose randomly k different points M_i $(i = 1, \ldots, k)$ in the data space; each M_i will serve as a prototype for a cluster C_i.
2. Assign each data element to the cluster with the closest prototype.
3. Recalculate each M_i as the mean of all elements of C_i.
4. Repeat steps 2 and 3 until the M_i and C_i no longer change.

Here step 2 is known as the *assignment step* and step 3 is known as the *update step*.

k-means always converges to a (possibly local) optimum. The proof of this (see, for instance [10]) involves the fact that the sum of all distances from one element to its cluster's prototype can only decrease in each update and assignment step, and only a finite number of such decrements is possible.

4 K-Means with the Hybrid Similarity Measure

K-means can in principle be used with all sorts of similarity measures; however, it also needs a center measure (e.g., the mean), and to guarantee convergence this center measure must be *compatible* with the similarity measure, that is, reassigning elements to clusters must lead to a monotonic increase (or decrease) of some aggregate function of the similarities beween elements and centers (e.g., increasing sum of similarities). We will call this aggregate function the *overall similarity*.

In our setting, the data elements are annotated vertices in a graph. This raises the question how to calculate the "prototypical vertex" of a subset of vertices from an annotated graph. If annotations are vectors, we can easily define the annotation of prototype M_i as the mean of all annotations $\lambda(v)$ where $v \in C_i$. But our hybrid similarity measures are based on $\mathcal{S}_{neighbor} : V \times V \to \mathbb{R}$, which also needs the prototype to be connected to nodes in the graph. Since this is not the case, we cannot compute the contextual similarity between the prototype and a data element.

We will discuss three ways around this problem. The first one is to use *k-medoids* instead of k-means; k-medoids always uses existing data elements as centers, which solves the above problem. An alternative is to define the center as an annotation, and define similarity between a node and an annotation, rather than between two nodes. Our third method will be a more efficient approximation of the second. We next discuss these three methods.

4.1 *K*-medoids

K-medoids [7] is similar to k-means, but differs from it in that the prototype of a cluster (in this case known as the medoid) is always an element from the data

set: in the beginning (step 1) it is a random data element; and during an update step (step 3), M_i becomes the element of C_i for which the overall similarity to all others elements of C_i is maximal.

It is easy to see that the hybrid similarity measure can be applied here without problems: since the prototype is always an element in the data set (i.e., a node in the graph), the similarity with other elements can be calculated. To compute the new prototype, one only needs to compute for each element the sum of the similarities to all other elements in that cluster, to determine which is the largest.

K-medoids can be a good alternative for k-means. It is known to be better than k-means when it comes to handling outliers [4], but more limited in its choice of prototypes [8], and less efficient when handling big data sets [4]. The latter is due to the fact that to calculate the new prototype of a cluster in k-medoids one needs to calculate the distance from every node in that cluster to every other node in that cluster, while for k-means one only needs to calculate the mean of all nodes in it.

4.2 *K*-means-NAM: *K*-means with Neighbor Annotation Means

The second solution we explore, is to define the center as an annotation instead of a node. Recall that the contextual similarity $S_{context}$ is a symmetrized version of $S_{neighbor}$. The definition of the latter (see (1)) uses for the first element (v) only its annotation $\lambda(v)$, not its location in the graph. Thus, the neighbor similarity $\mathcal{S}_{neighbor}$ can be rewritten as a function $\mathcal{S}'_{neighbor} : \mathcal{A} \times V \to \mathbb{R}$ that is defined by:

$$\mathcal{S}'_{neighbor}(M, v) = \frac{\sum_{w \in V} \mathcal{S}_{content}(M, \lambda(w)) \cdot \phi(w, v)}{\sum_{w \in V} \phi(w, v)} \tag{4}$$

We can use this asymmetric neighbor similarity instead of the contextual similarity as described in (2). Also the combined similarity can then be rewritten as a function $\mathcal{S}'_{combined} : \mathcal{A} \times V \to \mathbb{R}$ that is defined by:

$$\mathcal{S}'_{combined}(M, v) = c_1 \cdot \mathcal{S}_{content}(M, v) + c_2 \cdot \mathcal{S}'_{neighbor}(M, v) \tag{5}$$

In this case the new mean of a cluster can be calculated as the average of the annotations of all elements in that cluster, and it is still possible to calculate the similarity between a mean and an element from the data set.

This approach causes a new problem though: the proposed center measure and similarity measure are not compatible, and as a result, k-means may no longer converge. In the update step, the new prototype is the mean of the cluster element's annotations, which increases the average *content* similarity between M and the nodes, but not necessarily the *neighbor* similarity between M and these nodes. Consider an element v from the data set whose annotations of the neighbors differ a lot from its own annotation. When using contextual similarity, v will be placed in a cluster with a mean that is close to the annotations of the *neighbors* of v, but when updating the new mean for this cluster, this will be done using the annotation of v; this will pull the mean towards v's own annotation, and away from the annotation of its neighbors. The effect could be that the

new mean will be far from the annotations of v's neighbors, so the monotonic increase of the overall similarity is no longer guaranteed, and the algorithm may not converge.

To ensure convergence, we need to redefine the center measure to be compatible with the similarity measure. As described in Section 3, $\lambda : V \to \mathcal{A}$ assigns an annotation to a vertex. Now let $\lambda' : V \to \mathcal{A}$ be a new function that assigns another annotation to a vertex:

$$\lambda'(v) = \frac{\sum_{w \in V} \lambda(w) \cdot \phi(v, w)}{\sum_{w \in V} \phi(v, w)} \tag{6}$$

$\lambda'(v)$ is the mean annotation of all neighbors of v. It is easily seen that calculating the center as the average of these new annotations is compatible with the proposed similarity measure.

Following the same reasoning, when using the combined similarity instead of the contextual one, the annotation function $\lambda'' : V \to \mathcal{A}$ should be used:

$$\lambda''(v) = \frac{\lambda(v) + \lambda'(v)}{2} \tag{7}$$

This setup, which we call *k-means-NAM* (k-means with neighbor annotation means) is the second solution proposed to enable the use of a k-means-like algorithm with a hybrid similarity measure.

4.3 K-means-NAMA: *K*-means-NAM Efficiently Approximated

The solution proposed in Section 4.2 is less efficient than the original k-means algorithm. To calculate the similarity between a mean and an element v, k-means only needs to calculate one content similarity. k-means-NAM, on the other hand, needs to calculate the content similarity between the prototype and all neighbors of v (in order to compute their average), which makes it a number of times slower.

A more efficient alternative to this is to average out the neighbor annotations themselves, instead of averaging out the similarities. That is, with $v_1, \ldots, v_n$ the neighbors of v, instead of computing $\sum_i \mathcal{S}_{content}(M, \lambda(v_i))/n$, we could compute $\mathcal{S}_{content}(M, \sum_i \lambda(v_i)/n)) = \mathcal{S}_{content}(M, \lambda'(v))$. These two are mathematically different, and generally do not give the same outcome, but they approximate each other well when the v_i are "in the same direction" from a. The advantage of this additional approximation is that for each v, $\lambda'(v)$ can be computed once in advance, and substituted for $\lambda(v)$, after which the standard k-means algorithm can be run. In the same way that using λ' instead of λ allows us to approximate k-means-NAM with contextual distance, using λ'' approximates k-means-NAM with the combined distance. We call this approximation k-means-NAMA.

5 Experimental Results

To evaluate the usefulness of the proposed methods, a few questions need to be answered experimentally. K-medoids can be used both with contents-based

Table 1. Characteristics of the five subsets created from the Cora data set

DATA SET	CLASSES	PAPERS	EDGES	DENSITY
D_1	8	752	2526	3.36
D_2	17	1779	6658	3.74
D_3	24	2585	10754	4.16
D_4	31	4779	19866	4.17
D_5	45	8025	41376	5.16

or hybrid similarities; does the use of a hybrid similarity yield better results? For k-means, we have to choose between standard k-means with the content-based similarity, or an approximative variant that takes contextual information into account; does the use of contextual information compensate for a possible quality loss due to having to use an approximate method? Finally, how do the three contextual methods compare?

5.1 Experimental Setup

We evaluate the methods on the Cora dataset [11], an often-used benchmark. This set contains scientific papers divided into 70 categories. A paper can have multiple classes. For 37,000 of these papers, abstracts are available for keyword extraction, and the citations between papers are also available. In our context, papers are nodes in a graph, the abstracts are their annotations, and the citations between papers form the edges. We cluster the papers based on their annotations and citations. The quality measure for the clustering will relate to how well papers in the same cluster belong to the same classes (note that the classes themselves are not part of the annotation, only the abstracts are).

From Cora, five different subsets have been created. The first subset contains papers from 8 different classes. For every next subset, papers from several additional classes have been added. Only papers that have an abstract and are connected to (i.e., cite or are cited by) at least one other paper in the subset are regarded. Table 1 shows some of the characteristics of these subsets.

For every data set D_1 through D_5, V is defined by all papers in that data set. For all $v, w \in V$, $(v, w) \in E$ when v cites w or vice versa. Let W be the set of all keywords that can be found in the abstracts of all elements of all data sets D_i, and m the total number of keywords; $\mathcal{A} \subseteq \mathbb{R}^m$ and $\lambda(v)$ is the set of keywords that are in the abstract of v. For the content-based similarity between two elements $v, w \in V$ we use the Jaccard index [6]:

$$\mathcal{S}_{content} = \frac{|\lambda(v) \cap \lambda(w)|}{|\lambda(v) \cup \lambda(w)|}. \tag{8}$$

The three solutions as proposed in Section 4, have been used to cluster the elements in the 5 data sets as described in this section. For every combination this has been done for all three similarities (content-based, contextual and combined). The value for k was varied from 100 to 2. Keep in mind that k-means-NAM(A), used with the content-based similarity, is actually the regular k-means algorithm.

Table 2. Percentual difference in quality of the found clusters by k-means-NAMA compared to the found clusters by k-means-NAMA

	CONTEXTUAL SIMILARITY		COMBINED SIMILARITY	
DATA SET	ABSOLUTE DIFFERENCE	AVERAGE DIFFERENCE	ABSOLUTE DIFFERENCE	AVERAGE DIFFERENCE
D_1	2.4%	-1.2%	2.7%	-1.3%
D_2	2.4%	+1.0%	1.9%	-0.2%
D_3	1.6%	-0.7%	1.8%	+0.7%
D_4	0.9%	+0.2%	1.3%	-0.8%
D_5	1.3%	+0.4%	2.0%	+1.2%

The found clusterings were evaluated by looking at every pair of elements in a cluster and calculating the Jaccard index of their categories. In the Cora data set, papers can have multiple categories, hence, the Jaccard index is used to resolve partial matching. The average of all these Jaccard indices is the quality of the clustering.

The experiments have been done for $k = 100, 95, 90, \ldots, 20, 18, 16, \ldots, 2$. A set of experiments where a clustering is found once for every k is called a *series*. As k-means and k-medoids depend on the random initial choice of prototypes, every "series" has been done 100 times and 100 experiments with the same k on the same data set is called a *run*. Of these runs the average results are reported.

5.2 *K*-means-NAM vs *K*-means-NAMA

In Section 4.3 we hypothesized that k-means-NAMA would get similar results as k-means-NAM, but faster. This can be tested by regarding the percentual difference in score between the results of the runs with k-means-NAMA and the same runs with k-means-NAM. A positive percentage means that k-means-NAMA outperformed k-means-NAM, and a negative percentage the opposite. Also the absolute value of these percentages are taken into concern. Table 2 shows these results. First, the averages of all runs in a data set for the absolute value of these percentages are small, indicating there is not a lot of difference in performance. Second, when the sign of the percentual differences are also taken into account, the differences are even smaller, indicating that one is not overall better than the other. These conclusions hold for both the contextual and the combined similarity.

Table 3 shows the average computation time each method needed to finish one series of clusterings. With the content-based similarity, there is not much difference. This is as expected, since here, k-means-NAM boils down to regular k-means. For the contextual and combined similarities, however, there is a big difference: the time that it takes k-means-NAMA to complete a series remains about the same, while k-means-NAM needs about 4 times as long for the contextual similarity and about 5 times as long for the combined similarity.

Since there is no real difference in quality between k-means-NAM and k-means-NAMA, from now on we only consider the results for k-means-NAMA.

Table 3. CPU time, in seconds, for k-means-NAM and k-means-NAMA to do one series as defined in Section 5.1 (on a Intel® Quad Core™2, 2.4GHz with 4 Gb memory), for the content-based, contextual, and combined similarities

	CONTENT-BASED		CONTEXTUAL		COMBINED	
DATA SET	k-MEANS-NAM	k-MEANS-NAMA	k-MEANS-NAM	k-MEANS-NAMA	k-MEANS-NAM	k-MEANS-NAMA
D_1	$7.2 \cdot 10^1$	$8.6 \cdot 10^1$	$3.0 \cdot 10^2$	$1.1 \cdot 10^2$	$4.0 \cdot 10^2$	$1.1 \cdot 10^2$
D_2	$4.8 \cdot 10^2$	$5.9 \cdot 10^2$	$1.5 \cdot 10^3$	$6.0 \cdot 10^2$	$2.1 \cdot 10^3$	$5.8 \cdot 10^2$
D_3	$9.7 \cdot 10^2$	$1.2 \cdot 10^3$	$4.0 \cdot 10^3$	$1.3 \cdot 10^3$	$4.9 \cdot 10^3$	$1.3 \cdot 10^3$
D_4	$3.5 \cdot 10^3$	$4.2 \cdot 10^3$	$1.3 \cdot 10^4$	$4.7 \cdot 10^3$	$1.7 \cdot 10^4$	$4.4 \cdot 10^3$
D_5	$1.1 \cdot 10^4$	$1.3 \cdot 10^4$	$4.5 \cdot 10^4$	$1.6 \cdot 10^4$	$5.4 \cdot 10^4$	$1.4 \cdot 10^4$

5.3 Quality Improvement due to the Hybrid Similarity Measures

Figure 1 shows the results for clustering the subsets D_1 and D_4. The other subsets show similar characteristics. Table 4 shows the average results for all data sets for k-medoids and k-means-NAMA. On Cora, using the hybrid similarity measure indeed improves the quality of the found clustering, as compared to using the content-based similarity, for both k-medoids and k-means. There is no conclusive evidence, however, which one is best among k-medoids and (approximative) k-means.

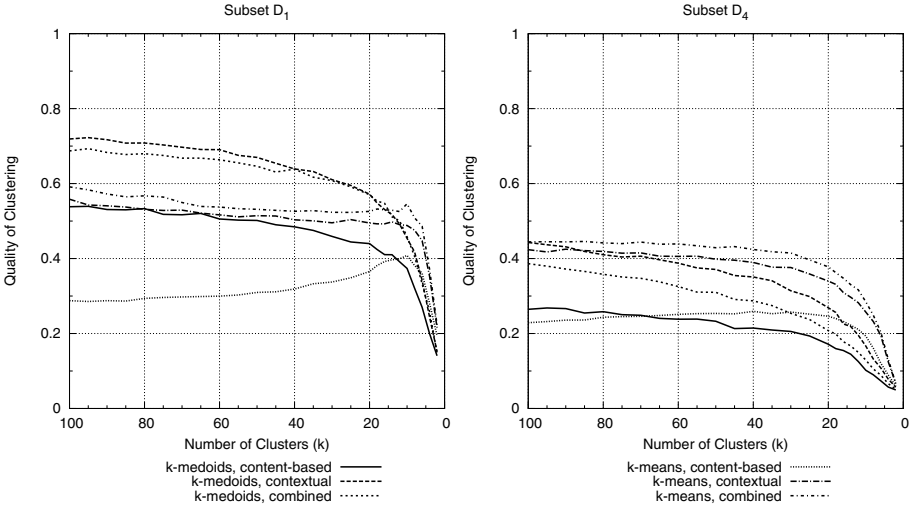


Fig. 1. Clustering quality for D_1 (left) and D_4 (right) for k-medoids and k-means-NAMA

Table 4. Average quality found and percentual improvement with regards to the content-based similarity, for k-medoids (upper half) and k-means-NAMA (lower half)

k-MEDOIDS					
	CONTENT-BASED	CONTEXTUAL		COMBINED	
DATA SET	QUALITY	QUALITY	IMPROVEMENT	QUALITY	IMPROVEMENT
D_1	0.44	0.57	+29%	0.58	+32%
D_2	0.25	0.36	+41%	0.39	+55%
D_3	0.19	0.29	+50%	0.33	+71%
D_4	0.19	0.25	+33%	0.30	+59%
D_5	0.14	0.19	+35%	0.24	+72%
k-MEANS-NAMA					
	CONTENT-BASED	CONTEXTUAL		COMBINED	
DATA SET	QUALITY	QUALITY	IMPROVEMENT	QUALITY	IMPROVEMENT
D_1	0.32	0.52	+60%	0.49	+51%
D_2	0.23	0.42	+79%	0.40	+71%
D_3	0.21	0.37	+77%	0.35	+70%
D_4	0.22	0.37	+68%	0.34	+55%
D_5	0.19	0.36	+74%	0.31	+62%

6 Conclusion

We have discussed how a hybrid similarity for nodes in a graph (taking into account both contents and context) can be used with k-means-like clustering methods. K-means cannot be employed in a straightforward way because the concept of a "mean node" cannot be defined; however, it can be approximated by k-medoids and by two newly proposed methods. These two methods boil down to using approximate similarity or center measures so that k-means becomes applicable. The main conclusions from this work are that: (1) k-means clustering can indeed work with hybrid similarities, if adapted appropriately; (2) the use of a hybrid similarity with (adapted) k-means or k-medoids does yield better clusters, compared to content-based similarities; (3) the adapted k-means approaches sometimes work better than k-medoids, so they are a valuable alternative to it but do not make it redundant.

Acknowledgements. The authors thank Walter Kosters for multiple suggestions w.r.t. formulation and writing style. This research is funded by the Dutch Science Foundation (NWO) through a VIDI grant.

References

1. Cook, D.J., Holder, L.B.: Substructure discovery using minimum description length and background knowledge. Journal of Artificial Intelligence Research 1, 231–255 (1994)
2. Flake, G.W., Tarjan, R.E., Tsioutsiouliklis, K.: Graph clustering and minimum cut trees. Internet Mathematics 1, 385–408 (2004)

3. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12), 7821–7826 (2002)
4. Han, J., Kamber, M., Tung, A.: Spatial clustering methods in data mining: A survey. In: Miller, H., Han, J. (eds.) Geographic Data Mining and Knowledge Discovery, Taylor & Francis, Taylor (2001)
5. Hartigan, J.: Clustering Algorithms. Wiley, Chichester (1975)
6. Jaccard, P.: Étude comparative de la distribution florale dans une portion des alpes et des jura. Bulletin del la Soci'etè Vaudoise des Sciences Naturelles 37, 547–579 (1901)
7. Kaufman, L., Rousseeuw, P.J.: Clustering by means of medoids. In: Dodge, Y. (ed.) Statistical Data Analysis Based on the L_1 Norm and Related Methods, pp. 405–416. Elsevier Science, Amsterdam (1987)
8. Kirsten, M., Wrobel, S.: Extending K-means clustering to first-order representations. In: Cussens, J., Frisch, A. (eds.) ILP 2000. LNCS (LNAI), vol. 1866, pp. 112–129. Springer, Heidelberg (2000)
9. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Berkeley (1967)
10. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
11. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Information Retrieval Journal 3, 127–163 (2000)
12. Neville, J., Adler, M., Jensen, D.: Clustering relational data using attribute and link information. In: Proceedings of the Text Mining and Link Analysis Workshop, Eighteenth International Joint Conference on Artificial Intelligence (2003)
13. Ramon, J.: Clustering and instance based learning in first order logic. PhD thesis, K.U.Leuven, Dept. of Computer Science (2002)
14. Sneath, P.H.A., Sokal, R.R.: Numerical Taxonomy - The Principles and Practice of Numerical Classification. W.H. Freeman, San Francisco (1973)
15. Witsenburg, T., Blockeel, H.: A method to extend existing document clustering procedures in order to include relational information. In: Kaski, S., Vishwanathan, S., Wrobel, S. (eds.) Proceedings of the Sixth International Workshop on Mining and Learning with Graphs, Helsinki, Finland (2008)
16. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. In: Proceedings of the VLDB Endowment, Lyon, France, pp. 718–729. VLDB Endowment (2009)

Pairwise Constraint Propagation for Graph-Based Semi-supervised Clustering

Tetsuya Yoshida

Graduate School of Information Science and Technology,
Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
yoshida@meme.hokudai.ac.jp

Abstract. This paper proposes an approach for pairwise constraint propagation in graph-based semi-supervised clustering. In our approach, the entire dataset is represented as an edge-weighted graph by mapping each data instance as a vertex and connecting the instances by edges with their similarities. Based on the pairwise constraints, the graph is then modified by contraction in graph theory to reflect must-link constraints, and graph Laplacian in spectral graph theory to reflect cannot-link constraints. However, the latter was not effectively utilized in previous approaches. We propose to propagate pairwise constraints to other pairs of instances over the graph by defining a novel label matrix and utilizing it as a regularization term. The proposed approach is evaluated over several real world datasets, and compared with previous regularized spectral clustering and other methods. The results are encouraging and show that it is worthwhile to pursue the proposed approach.

1 Introduction

We have proposed a graph-based approach for semi-supervised clustering [9]. In our approach the entire dataset is represented as an edge-weighted graph by mapping each data instance as a vertex and connecting the instances by edges with their similarities. The graph is then modified by contraction in graph theory [3] to reflect must-link constraints, and graph Laplacian in spectral graph theory [7] is utilized to reflect cannot-link constraints. However, the latter was not effectively utilized in previous approach, and performance improvement with constraints remained rather marginal.

We propose a method to remedy the above problem by propagating pairwise constraints to other pairs of instances over the graph. We define a novel label matrix based on the specified constraints and utilizing it as a regularization term. Although the proposed approach utilizes graph Laplacian [1], it differs since pairwise constraints are utilized and propagated to conduct semi-supervised clustering. The proposed approach is evaluated over several real world datasets, and compared with previous regularized spectral clustering and other methods. The results are encouraging and indicate the effectiveness of the proposed approach to exploit the information available in pairwise constraints.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 358–364, 2011.

2 Semi-Supervised Clustering

2.1 Preliminaries

We use an italic upper letter to denote a set. For a finite set X, $|X|$ represents its cardinality. We use a bold italic lower letter to denote a vector, and a bold normal upper letter to denote a matrix. In this paper, we assume that the edge weights in a graph is non-negative and can be represented as $\mathbf{W}$.

2.2 Problem Setting

Based on previous work [8,6,5], we consider two kinds of constraints called must-link constraints and cannot-link constraints. When a pair of instances (i, j) is included in must-link constraints C_{ML}, the instances are to be in the same cluster; on the other hand, if (i, j) is included in cannot-link constraints C_{CL}, these are to be in different clusters. Hereafter, these constraints are also called as must-links and cannot-links, respectively.

3 Pairwise Constraint Propagation over Graph

3.1 Graph-Based Semi-supervised Clustering

We have proposed a graph-based approach for semi-supervised clustering [9]. In our approach, two kinds of constraints are treated as edge-labels. In our approach a set of vertices connected by must-links are contracted into a vertex based on graph contraction in graph theory [3], and weights are re-defined over the contracted graph. Next, weights on the contracted graph are further modified based on cannot-links. Finally, a projected representation of the given data is constructed based on the pairwise relation between instances, and clustering is conducted over the projected representation.

3.2 Weight Modification via Cannot-Links

In our previous approach [9], weights $\mathbf{W}^{'}$ on the contracted graph G' are discounted based on cannot-links, and another graph G'' is constructed. Weights $\mathbf{W}^{''}$ on G'' are defined as

$$\mathbf{W}^{''} = (\mathbf{1}_{n'}\mathbf{1}_{n'}^{t} - \nu\mathbf{C}_{CL}) \odot \mathbf{W}^{'} \tag{1}$$

where n' is the number of vertices in G', $\mathbf{1}_{n'} = (1,\ldots,1)^t$, $\mathbf{C}_{CL}$ is a binary matrix where only the elements for cannot-links are set to 1, $\odot$ stands for the Hadamard product of two matrices, and $\nu \in [0, 1]$ is a discounting parameter.

However, only the weights on cannot-links are modified in eq.(1). Thus, it was insufficient to exploit the information available from cannot-links, and the performance improvement was rather marginal. We propose a method to remedy this problem in the following section.

3.3 Regularization via Cannot-Links

When constructing the projected representation, by reflecting cannot-links, we propose the following objective function based on cannot-links:

$$J_2 = tr(\mathbf{H}^t\mathbf{L}\mathbf{H}) + \lambda\ tr(\mathbf{H}^t\mathbf{S}\mathbf{H}) \quad (2)$$
$$s.t.\ \ \mathbf{H}^t\mathbf{D}\mathbf{H} = \mathbf{I}$$

Here, the matrix $\mathbf{L}$ corresponds to the graph Laplacian of the contracted graph G'. The matrix $\mathbf{S}$ is defined based on cannot-links, and explained below. The real number $\lambda \geq 0$ is a regularization parameter. The second term in eq. (2) corresponds to a regularization term.

Utilization of pairwise constraints as the regularization term was also pursued [2,4]D In previous approaches, the matrix $\mathbf{S}$ was usually defined as:

$$s_{ij} = \begin{cases} -1 & if \quad (i,j) \in C_{ML} \\ 1 & if \quad (i,j) \in C_{CL} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

However, only the pairs of instances explicitly specified in pairwise constraints were considered. Since other pairs of instances are not utilized for regularization, specified constraints are not effectively utilized for semi-supervised clustering.

To cope with the above problem, we propose to propagate pairwise constraints to other pairs of instances through the following matrix $\mathbf{S}$. Note that the proposed matrix $\mathbf{S}$ is defined based on cannot-links in our approach.

For a pair of vertices (instances) $(i,j) \in C_{CL}$, suppose a vertex k is adjacent to the vertex j and $(i,k) \notin C_{CL}$. In this case, we try to utilize the weights $w'_{ik}, w'_{jk} \in [0,1]$ on the edges adjacent to the vertex k in the contracted graph G', and set $s_{ik} = (1 - w'_{ik})w'_{jk}$. On the other hand, when $w'_{jk} = 0$ (*i.e.*, the vertices j and k are not similar at all in G'), since s_{ik} is set to 0, the constraint on the pair of instances (i,j) is not propagated to the pair (i,k), and the latter is not utilized for regularization.

We aggregate the influence of multiple constraints by their average and define the matrix $\mathbf{S}$ as follows:

$$s_{ik} = \begin{cases} 1 \\ \frac{1}{m_{ik}} \left\{ \sum_{(i,j)\in C_{CL}} (1 - w'_{ik})w'_{jk} + \sum_{(k,j)\in C_{CL}} (1 - w'_{ik})w'_{ji} \right\} \\ 0 \end{cases} \quad (4)$$

where m_{ik} represents the number of constraints which are considered for the summation. The first line in eq. (4) is for when $(i,k) \in C_{CL}$. The second line in eq. (4) is for when $((i,j) \in C_{CL}) \vee ((k,j) \in C_{CL})$ for $(x_i, x_j) \in C_{CL}$. This corresponds to the propagation of constraints to the neighboring pairs. When these two conditions are not satisfied, the third line is utilized. By utilizing the matrix $\mathbf{S}$ in eq. (4), cannot-links are propagated to the neighboring pairs of instances and the propagated ones are also utilized for regularization in eq. (2).

4 Evaluations

4.1 Experimental Settings

Datasets. Based on previous work [6], we conducted evaluations on 20 Newsgroup (20NG) [1]. These datasets contain documents and their cluster labels. For 20NG, we created three sets of groups. As in [6], 50 documents were sampled from each group in order to create one dataset, and 10 datasets were created for each set of groups. For each dataset, we conducted stemming using porter stemmer and MontyTagger removed stop words, and selected 2,000 words with large mutual information. We conducted experiments on the TREC datasets, however, results on other datasets are omitted due to page limit.

Evaluation Measure. For each dataset, cluster assignment was evaluated w.r.t. the following Normalized Mutual Information (NMI). Let T, $\hat{T}$ stand for the random variables over the true and assigned clusters. NMI is defined as $NMI = 2I(\hat{T};T)/(H(\hat{T}) + H(T))$ where $H(T)$ is Shannon Entropy. The larger NMI is, the better the result is. All the methods were implemented with R.

Comparison. We compared the proposed pGBSSC with GBSSC [9], regularized spectral clustering in eq.(2) with $\mathbf{S}$ in eq.(3), SCREEN [6], and PCP [5].

Parameters. The parameters are: 1) the number of constraints, 2) the pairs of instances specified as constraints. Following [6,5], pairs of instances were randomly sampled from a dataset to generate pairwise constraints. We set the number of constraints for must-links and cannot-links as equal (i.e., $|C_{ML}| = |C_{CL}|$), and varied the number of constraints to investigate their effects.

Representation. each data instance $\boldsymbol{x}$ was normalized as $\boldsymbol{x}^t\boldsymbol{x} = 1$, and Euclidian distance was utilized for SCREEN as in [6]. Cosine similarity, which is widely utilized as the standard similarity measure in document processing, to define the weights on the graph. The number of dimensions of the subspace was set to the number of clusters in each dataset. Following the procedure in [5], a nearest neighbor graph was constructed for PCP in each dataset by setting the number of neighboring vertices to 10.

Evaluation Procedure. Clustering with the same number of constraints was repeated 10 times with different initial configuration in clustering. In addition, the above process was also repeated 10 times for each number of constraints. Thus, the average of 100 runs is reported in the following section. *NMI* was calculated based on the ground-truth label in each dataset.

4.2 Evaluation of Regularization Based on Cannot-Links

The proposed approach utilizes cannot-links as a regularization term (the second term in eq. (2)), and controls its influence with the parameter λ. We varied parameter values and investigated their effects. The results are shown in Fig. 1. The horizontal axis in Fig. 1 corresponds to the value of ν in eq.(1) and λ in

[1] http://people.csail.mit.edu/~jrennie/20Newsgroups/

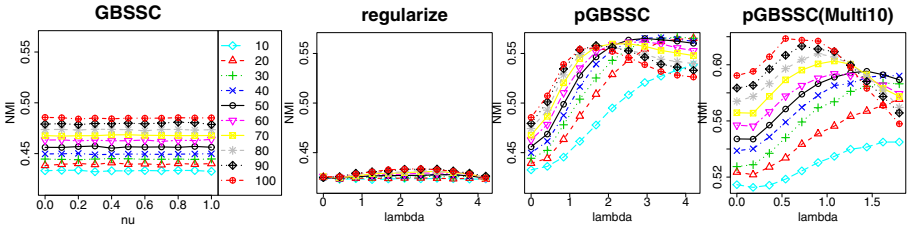


Fig. 1. Influence of λ in eq. (2) (from left, 1st for GBSSC [9], 2nd: with matrix $\mathbf{S}$ in eq. (3), 3rd and 4th: with $\mathbf{S}$ in eq. (4). 1st to 3rd for Multi15, 4th for Multi10.)

eq.(2). The number in the legend is the number of constraints (*i.e.*, $|C_{CL}|$). In Fig. 1, from left, the 1st figure corresponds to our previous method GBSSC [9], 2nd to the previous regularized spectral clustering with the matrix $\mathbf{S}$ in eq. (3), and 3rd and 4th to our proposal ($\mathbf{S}$ in eq. (4)). The 1st to 3rd figures are the results for Multi15, and 4th is for Multi10.

As shown in Fig. 1, in our previous GBSSC (1st figure), the value of ν does not matter. Thus, it could not exploit the information contained in cannot-links effectively. In the previous regularized spectral clustering with the matrix $\mathbf{S}$ in eq. (3), almost no difference was observed with respect to the number of constraints and the value of λ (2nd figure in Fig. 1). On the other hand, the proposed approach showed the improvement of *NMI* with respect to the value of λ (3rd and 4th figures in Fig. 1).

A cannot-link on a pair of instances contributes to separating two clusters to which the instances are assigned, but the possible number of combinations of clusters depend on the number of clusters. Thus, contribution of each cannot-link can be relatively small when the number of possible combinations is large. In the following experiments, the parameter λ was set as $\lambda = \lambda_0 \cdot {}_kC_2$ (here, k stands for the number of clusters, ${}_kC_2$ represents the number of possible combinations of clusters), and λ_0 was set to 0.02 based on our preliminary experiments.

4.3 Evaluation on Real World Datasets

In the following figures, the horizontal axis corresponds to the number of constraints; the vertical one corresponds to *NMI*. In the legend, red bold lines correspond to the proposed pGBSSC, pink solid lines to GBSSC, blue dot-dash lines (regularize) to regularized spectral clustering with $\mathbf{S}$ in eq. (3), black dotted lines to SCREEN, and green dashed lines to PCP.

Fig. 2 shows that the proposed pGBSSC outperformed other methods in most datasets. Especially, pGBSSC was much better than other methods in Multi10, Multi15. Compared with previous GBSSC and regularize (with $\mathbf{S}$ in eq. (3)), the proposed pGBSSC showed more performance improvement by utilizing the pairwise constraints in these datasets. On the other hand, the performance of

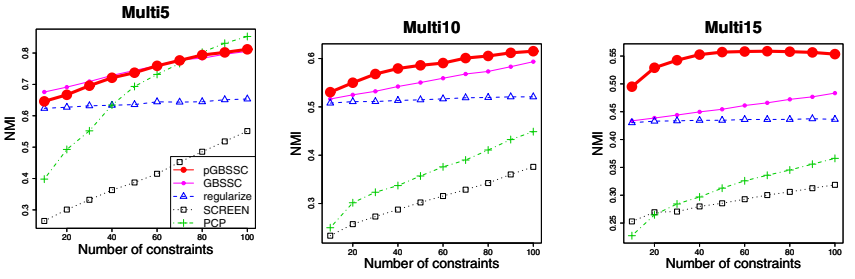


Fig. 2. Results on real-world datasets

pGBSSC and GBSSC was almost the same in other datasets. In addition, although PCP does not seem effective with small number of constraints, it showed large performance gain as the number of constraints increased.

5 Concluding Remarks

This paper proposes an approach for pairwise constraint propagation in semi-supervised clustering. In our approach, the entire data is represented as an edge-weighted graph with the pairwise similarities among instances. Based on the pairwise constraints, the graph is modified by contraction in graph theory and graph Laplacian in spectral graph theory. However, the latter was not effectively utilized in previous approaches. In this paper we proposed to propagate pairwise constraints to other pairs of instances over the graph by defining a novel label matrix and utilizing it as a regularization term. The proposed approach was evaluated over several real world datasets and compared with previous regularized spectral clustering and other methods. The results are encouraging and show that it is worthwhile to pursue the proposed approach. Especially, propagated constraints contributed to further improve performance in semi-supervised clustering.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2002)
2. Bie, T.D., Suykens, J., Moor, B.D.: Learning from General Label Constraints. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 671–679. Springer, Heidelberg (2004)
3. Diestel, R.: Graph Theory. Springer, Heidelberg (2006)
4. Goldberg, A.B., Zhu, X., Wright, S.: Dissimilarity in graph-based semi-supervised classification. In: Proc. of AISTAT 2007, pp. 155–162 (2007)
5. Li, Z., Liu, J., Tang, X.: Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In: ICML 2008, pp. 576–583 (2008)
6. Tang, W., Xiong, H., Zhong, S., Wu, J.: Enhancing semi-supervised clustering: A feature projection perspective. In: Proc. of KDD 2007, pp. 707–716 (2007)

7. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
8. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: ICML 2001, pp. 577–584 (2001)
9. Yoshida, T., Okatani, K.: A Graph-Based Projection Approach for Semi-supervised Clustering. In: Kang, B.-H., Richards, D. (eds.) PKAW 2010. LNCS(LNAI), vol. 6232, pp. 1–13. Springer, Heidelberg (2010)

Space-Time Roll-up and Drill-down into Geo-Trend Stream Cubes

Anna Ciampi[1], Annalisa Appice[1], Donato Malerba[1], and Angelo Muolo[2]

[1]Department of Computer Science, University of Bari "Aldo Moro"
via Orabona, 4 - 70126 Bari - Italy
[2]Rodonea®s.r.l., via Fiume, 8 Monopoli, Bari - Italy
{aciampi,appice,malerba}@di.uniba.it, angelo.muolo@rodonea.com

Abstract. We define a new kind of stream cube, called *geo-trend stream cube*, which uses trends to aggregate a numeric measure which is streamed by a sensor network and is organized around space and time dimensions. We specify space-time roll-up and drill-down to explore trends at coarse grained and inner grained hierarchical view.

1 Introduction

A traditional data cube is constructed as a multi-dimensional view of aggregates computed in static, pre-integrated, historical data. Most of today emerging applications produce data which is continuously measured at a rapid rate, which dynamically changes and, hence, which is impossible to store entirely in a persistent storage device. Thus, the pervasive ubiquity of stream environments and the consolidated data cube technology have paved the way for the recent advances toward the development of a mature stream cube technology [2,7,3,5].

In this work, we extend the cube technology to the spatially distributed data streams of sensor networking. We consider sets of observations, called *snapshots*, transmitted for a numeric measure from a sensor network. Snapshots are transmitted, equally spaced in time, to a central server. This scenario poses the issues of any other streaming environment, i.e., the limited amount of memory, the finite computing power of the servers and, as an additional degree of complexity, the spatial autocorrelation of the measure thought the space of each snapshot.

As observed in [6], time and space are natural aggregation dimensions for the snapshots of a stream. This consideration paves the way to make a step forward in data warehouse research by extending the cube perspective towards a trend based cube storage of the snapshots in a stream. This new kind of cube, called *geo-trend stream cube*, organizes the stream storage around time and space dimensions. Trend polylines are computed and stored as cells of a persistent cube. Roll-up and drill-down operations are also supported to navigate the cube and display the trends at a coarse-grained/inner grained hierarchical view.

The paper is organized as follows. In the next Section, we introduce basic concepts. In Section 3, we present a geo-trend stream cuber called GeoTube and the space-time roll-up and drill-down supported operations. Finally, we illustrate an application of GeoTube.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 365–375, 2011.

2 Basic Concepts and Problem Definition

In this Section, we introduce basic concepts on streams, snapshots and window model, we describe the trend clusters and we define the geo-trend stream cube.

Snapshot Stream. A *snapshot stream* D is defined by the triple $(M, T, space(T))$, where: (1) M is a numeric measure; (2) T is the unbounded sequence of discrete, equally spaced time points; and (3) $space\colon T \mapsto \mathbb{R} \times \mathbb{R}$ is the function which maps a time point t into the finite set of 2D points which represent the spatial position (e.g., by latitude and longitude) of sensors measuring a value of M at t. As we consider sensors which do not move thought the space, a sensor is identified by its point position on the Earth. Nevertheless, the number of transmitting sensors may change in time as a sensor may pass from on to off (or vice-versa). A *snapshot* $D[t]$ is the one-to-one mapping between a time point $t \in T$ and the set of values streamed for M in D at t from the sensor points of $space(t)$.

In stream mining, several algorithms are defined to mine knowledge from streams [4]. Most of these algorithms segment the stream into consecutive windows, such that the stream can be efficiently processed window-by-window. Here the *count-based window model* is tailored to the scope of segmenting a snapshot stream. Let w be a window size, T is broken into consecutive windows (denoted as $window_w(T)$) such that a time window $W \in window_w(T)$ collects w consecutive time points of T. This segmentation of T implicitly defines a window segmentation of the stream D (denoted as $window_w(D)$). Let $W \in window_w(T)$ be a time window, the snapshot window $D[W] \in window_w(D)$ is the sub-series of w consecutive snapshots streamed in D at time points of W. $D[W]$ is represented by the triple $(M, W, space[W])$, which expresses that M is measured from sensors in $space[W]$ along the W time horizon. $space[W]$ is the finite set of sensor points which transmitted at least one value along W (i.e. $space[W] = \bigcup_{t \in W} space(t)$).

The motivation of resorting to a window based segmentation of the stream in this work is that snapshots naturally enable the computation of aggregate operators spread over the space, but processing windows of snapshots is the natural way to add a time extent to the computation of space aggregates.

Trend Cluster based Aggregation. The traditional aggregate operators are sum, avg, median and count. They are computed without paying attention to how measurements are arranged in space and/or in time. Differently, we consider a space-time aggregate operator which computes space-time aggregate, named trend clusters, from the snapshots of a window. Formally, a *trend cluster* TC_w^δ with window size w and trend similarity threshold δ is the triple (W, P, C) where (1) W is a w-sized time window ($W \in window_w(T)$); (2) P is a trend prototype represented as a time-series of w values (trend points) timestamped at the consecutive time points falling in W; and (3) C is a cluster of spatially close sensors whose measurements, transmitted along W, differ at worst δ from the corresponding trend points in P.

The system which currently computes trend clusters in a snapshot stream is SUMATRA. The system inputs a window size w, a trend similarity threshold δ

and the definition of spatial closeness relation between sensors (e.g., a distance relation such as nearby or far away, a directional relation such as north of). Then SUMATRA processes each w-sized window $D[W]$ of the snapshot stream D and discovers a set of trend clusters in $D[W]$. For each trend cluster, SUMATRA computes the cluster and trend simultaneously. In particular, the trend points are obtained as the median of the clustered values at each time points of the window. We observe that the trend based clusters discovered in $D[W]$ also define a δ depending spatial segmentation of $space[W]$ A more detailed description of SUMATRA is out of the scope of this paper, but it can be found in [1].

In this work, the use of the trend clusters to aggregate snapshots is inspired by the fact that knowing how "spatially clustered" measurements evolve in time draw useful conclusions. For example, in a weather system, it is information bearing how temperatures increase and/or decrease over regions of the Earth and how the shape of these regions change in time. Additionally, the trend-based visualization of snapshots can be considered close to the human interpretation.

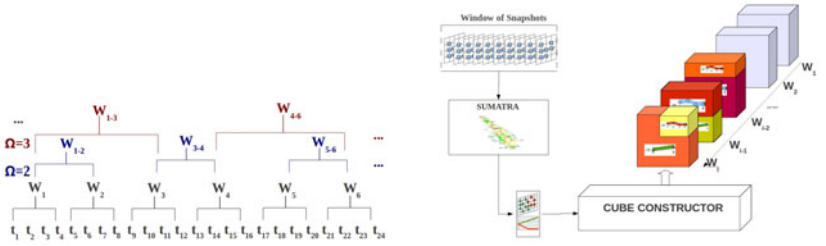
Geo-Trend Stream Cube. In data warehousing, a cube is defined by the triple (M, A, F) where M is the measure, A is the non empty set of dimensional attributes and F is the fact table. Based on this definition, a *geo-trend stream cube* $\mathcal{C}$ is defined as a special kind of cube where M is the measure streamed by a snapshot stream D, A comprises the stream time T and the stream space function $space(T)$ and F is the unbounded collection of timestamped snapshots from D. Each snapshot can be represented in F by a set of triple tuples $(t, [x, y], m)$ with t a time point of T, $[x, y]$ a sensor point of $space(t)$, and m a measurement of M. Each dimensional attribute is associated to a hierarchy of levels such that each level is a set of dimension values, and there exists a partial order based on a containment relation ($\sqsupseteq$) according to for two levels in a dimension the values of the higher level contain the values of the lower level. The time hierarchy, denoted as H_T, is uniquely defined by the size w of the window segmentation of T. The space hierarchy, denoted as $H_{space(T)}$, is time-dependently defined, so that for each window $W \in window_w(T)$, $H_{space[W]}$ is defined by trend similarity threshold δ used to discover a trend cluster segmentation of $D[W]$. The structure of both hierarchies is described below.

The Time Hierarchy H_T. Let w be the window size associated to $window_w(T)$. The hierarchy H_T is defined, depending on w, such that the containment relation:

$$\underbrace{T}_{time\ line} \sqsupseteq \underbrace{\{W^{\Omega}\}_{\Omega}}_{window\ of\ windows} \sqsupseteq \underbrace{W}_{window} \sqsupseteq \underbrace{t}_{time\ point}$$

is satisfied. t is a time point of T. W is a window of $window_w(T)$ and W^{Ω} is an higher level window which comprises Ω consecutive windows of $window_w(T)$. By varying Ω, alternative (xor) equal depth window of windows levels are defined in H_T (see Figure 1(a)).

The Space Hierarchy $H_{space(T)}$. Let δ be the trend similarity threshold. As the space segmentation associated to the set of trend clusters discovered with δ may change at each new window of the stream, the space hierarchy $H_{space(T)}$ changes window-by-window. Hence, $H_{space(T)}$ denotes an unbounded sequence



(a) The Time Hierarchy (b) The Architecture of GeoTube

Fig. 1. The time hierarchy H_T defined with $w = 4$. Ω varies between 2 and $\propto$, thus defining $\propto$ alternative (xor) equal depth levels in H_T (Figure 1(a)) and the architecture of the Geo-Trend Stream Cube (Figure 1(b)).

of hierarchies, that is, $H_{space(T)} = \{H_{space[W]}\}_{W \in window_w(T)}$, with $H_{space[W]}$ the space hierarchy for the horizon time of the window W. In particular, $H_{space[W]}$ is defined over $space[W]$ depending on δ such that the containment relation:

$$\underbrace{space[W]}_{space} \supseteq \underbrace{\{C^{\Delta}\}_{\Delta}}_{cluster\ of\ clusters} \supseteq \underbrace{C}_{cluster} \supseteq \underbrace{[x,y]}_{spatial\ point}$$

is satisfied. $[x, y]$ is a spatial point of $space[W]$. C is a cluster of spatially close sensors which transmit values whose trend polylines differ at worst δ from the trend prototype associated to C. C^{Δ} is an higher level cluster which groups spatially close clusters whose trend prototypes are similar along the window time with trend similarity threshold Δ. Two clusters are spatially close if they contain two spatially close sensor points. Also in this case, by varying Δ, we define alternative (xor) equal depth cluster of cluster levels into the hierarchy.

3 Geo-Trend Stream Cube

In the following we first present the algorithm to feed a geo-trend stream cube and then we illustrate drill-down/roll-up navigation of a geo-trend stream cube.

Cube Construction. GeoTube (GEO-Trend stream cUBEr) is designed as the component of a stream management system which is in charge of constructing on-line the geo-trend stream cube of a snapshot stream. In GeoTube, the request:

`CREATE GEOTRENDCUBE` $\mathcal{C}$ `WITH MEASURE` M `FROM STREAM` D
`GROUPING BY SPACE, TIME HAVING SpaceSimilarity` δ `AND Window` w

triggers the construction of the geo-trend stream cube $\mathcal{C}$ from the base stream $D(M, T, space(T))$. The cube $\mathcal{C}$ will be made permanent into a database by

considering windows as aggregation level for the time and spatial clusters as aggregation level for the space. The architecture of the GeoTube cuber component, which answers this request, is reported in Figure 1(b) and comprises: (1) a *snapshot buffer* which consumes snapshots as they arrive and pours them window-by-window into SUMATRA; (2) the system *SUMATRA* which is in charge of the trend cluster discovery process; and (3) the *cube slice constructor* which builds a new window slice of $\mathcal{C}$ from the trend clusters discovered by SUMATRA.

The definition of a window slice in geo-trend stream cube is coherent with the concept of slice formulated in the traditional cube technology. In particular, a window slice is the subset of the cube with the window as specific value on the time dimension. This means that, in GeoTube, a cube $\mathcal{C}$ is incrementally built each time a new snapshot window $D[W]$ is completed in the stream and by computing the associated window slice $\mathcal{C}[W]$ and adding this new slice to $\mathcal{C}$ along the time line T. The slice construction proceeds as follows. Let $TC[W]$ be the set of trend clusters that SUMATRA discovers in the buffered $D[W]$ with window size w and trend similarity threshold δ. The cube constructor inputs $TC[W]$ and uses this set to feed the new window slice $\mathcal{C}[W]$. This slice is built by assigning W as fixed value to the time dimension and by permitting the space dimension $space[W]$ to vary over the set of clusters included into $TC[W]$. Formally, $\mathcal{C}[W]$ will be completely defined by $TC[W]$ as follows:

$$\mathcal{C}[W] = \begin{array}{|c|c|}\hline \text{space} & \text{measure} \\ \hline C_1 & P_1 \\ \hline \ldots & \ldots \\ \hline C_q & P_q \\ \hline \end{array}$$

with $TC[W] = \{(W, C_k, P_k)\}_{k=1,2,\ldots,q}$. The time points falling in W are inserted as distinct points into the time hierarchy H_T and the window W is inserted, as grouping value, at the window level of H_T. Similarly, the spatial points of $space[W]$ feed a newly defined space hierarchy $H_{space[W]}$ and each cluster C_k of $TC[W]$ is inserted a distinct grouping value at the cluster level of $H_{space[W]}$. Finally, each trend prototype P_k is stored into the cube cell of $\mathcal{C}[W]$ associated to the cluster C_k. It is noteworthy that, due to the stream compression naturally operated by trend clusters [1], the memory size of C grows indefinitely, but less than memory size of the stream D.

Space-Time ROLL-UP and DRILL-DOWN. GeoTube provides the user with operators of roll-up and drill-down which are both defined in a space-time environment to permit the multi-level navigation of the cube by changing the abstraction levels of its dimensional attributes.

Space-Time ROLL-UP. The roll-up request is formulated as follows:

`ROLL-UP on` M `OF GEOTRENDCUBE` $\mathcal{C}$ `WITH SPACE` Δ `and TIME` Ω

and triggers the roll-up process to build a new cube $\mathcal{C}'$ from $\mathcal{C}$. The cube $\mathcal{C}'$ has the same base stream of $\mathcal{C}$. The time hierarchy H'_T collects the time points of T grouped, at the window level, into the windows of the $(w \times \Omega)$-sized segmentation of T. For each window $W' \in window_{w \times \Omega}(T)$, the space hierarchy $H'_{space[W']}$

collects the spatial points of $space[W']$. At the cluster level of $H'_{space[W']}$, the spatial points are grouped as the clusters of a trend cluster segmentation of $D[W']$ having a trend similarity upper bound $\Delta + \delta$.

The roll-up process to compute $\mathcal{C}'$ is iterative. At each iteration i, it pours the i-th series of Ω window slices of $\mathcal{C}$, denoted as $\langle \mathcal{C}[W_{(i-1)\Omega+1}], \mathcal{C}[W_{(i-1)\Omega+2}], \ldots, \mathcal{C}[W_{(i-1)\Omega+\Omega}] \rangle$, into a buffer synopsis. This series is processed to build a new window slice $\mathcal{C}'[W'_i]$. The process is iterated at each new i series of Ω window slices buffered from $\mathcal{C}$.

The construction of the slice $\mathcal{C}'[W'_i]$ proceeds as follows. Firstly, for each window slice $\mathcal{C}[W_{(i-1)\Omega+j}]$ (with j ranging between 1 and Ω), the associated trend cluster set $TC_{(i-1)\Omega+j} = \{(W_{(i-1)\Omega+j}, C_{(i-1)\Omega+j_k}, P_{(i-1)\Omega+j_k})\}_k$ is easily obtained by selecting (1) each value $C_{(i-1)\Omega+j_k}$ which is included in the cluster level of $H_{space[W_{(i-1)\Omega+j}]}$ and (2) each trend aggregate $P_{(i-1)\Omega+j_k}$ which is stored into the cell of $\mathcal{C}[W_{(i-1)\Omega+j}]$ in correspondence of the spatial cluster $C_{(i-1)\Omega+j_k}$. Then the series of trend cluster sets $\langle TC_{(i-1)\Omega+1}, TC_{(i-1)\Omega+2}, \ldots, TC_{(i-1)\Omega+\Omega} \rangle$ is processed and transformed into a new trend cluster set $TC'_i = \{(W'_i, C'_h, P'_h)\}_h$ such that: (1) the window $W'_i = \bigcup_{j=1}^{\Omega} W_{(i-1)\Omega+j}$ is a the window of T with size $w \times \Omega$; (2) the set $\{C'_h\}_h$ is a segmentation of $space[W'_i]$ into spatial clusters of sensors transmitting series of values along W'_i which differ at worst $\delta + \Delta$ from the associated trend prototype in $\{P'_h\}_h$; and (3) the set $\{P'_h\}_h$ is a set of trend prototypes along the time horizon of W'_i.

The trend cluster set TC'_i defines the new slice $\mathcal{C}'[W'_i]$ as follows. The time points of W'_i feed H'_T and are grouped into the window W'_i at the window level of H'_T; the spatial points of $space[W'_i]$ feed $H'_{space(W'_i)}$ and are grouped into the associated C'_h at the cluster level. Finally, each trend aggregate P'_h is stored into the cell of $\mathcal{C}'[W'_i]$ associated to the cluster C'_h.

The algorithm to construct TC'_i is reported in Algorithm 1. The algorithm is two stepped. In the first step (`SPACE ROLL-UP`), each input trend cluster set $TC_{(i-1)\Omega+j}$ is processed separately by navigating the space-hierarchy $H_{space(W_{(i-1)\Omega+j})}$ one level up according to Δ. This roll-up in space is performed by applying SUMATRA to the collection of trend prototypes collected into $TC_{(i-1)\Omega+j}$. SUMATRA is run with trend similarity threshold Δ by dealing each cluster of $TC_{(i-1)\Omega+j}$ as a single sensor and the associated trend prototype as the series of measurements transmitted by the cluster source. By means of SUMATRA, a set of trend "clusters of clusters", denoted as $\widetilde{TC}_{(i-1)\Omega+j}$, is computed under the assumption that two cluster sources are spatially close if they contain at least two sensors which are spatially close (lines 1-4 in the Main routine of Algorithm 1). We observe that the output $\widetilde{TC}_{(i-1)\Omega+j}$ is, once again, a segmentation of $D[W_{(i-1)\Omega+j}]$ into trend clusters with an upper bound for the trend similarity threshold which can be estimated under $\delta + \Delta$. In the second step (`TIME ROLL-UP`), the series of computed trend cluster sets $\langle \widetilde{TC}_{(i-1)\Omega+1}, \widetilde{TC}_{(i-1)\Omega+2}, \ldots, \widetilde{TC}_{(i-1)\Omega+\Omega} \rangle$ is now processed to compute the output trend cluster set TC'_i. We base this computation on the consideration that the sen-

sor points, which are repeatedly grouped together in a cluster for each window $W_{(i-1)\Omega+j}$ (with j ranging between 1 and Ω), represent measurements which evolve with a similar trend prototype along W'_i. This way, the construction of each new trend cluster (W'_i, C'_k, P'_k) starts by choosing the sensor $s \in space[W'_i]$ not yet clustered (line 7 in Main routine of Algorithm 1). Let $(W_{(i-1)\Omega+j}, \widetilde{C}^{[s]}_{(i-1)\Omega+j}, \widetilde{P}^{[s]}_{(i-1)\Omega+j}) \in \widetilde{TC}_{(i-1)\Omega+j}$ be the trend cluster of $\widetilde{TC}_{(i-1)\Omega+j}$ which contains s for the window $W_{(i-1)\Omega+j}$, (i.e. $s \in \widetilde{C}^{[s]}_{(i-1)\Omega+j}$) (lines 9-10 in Main routine of Algorithm 1). The new cluster C'_k initially contains s (line 12 in Main routine of Algorithm 1). Then it is expanded by grouping the spatially close sensors which are clustered at the same way of s along the series of windows $\langle W_{(i-1)\Omega+1}, W_{(i-1)\Omega+2}, \ldots,$ and $W_{(i-1)\Omega+\Omega}\rangle$ (line 13 in Main routine of Algorithm 1). In particular, the cluster expansion (see *expandCluster*() into Algorithm 1) is performed by adding the unclustered sensor points $t \in space[W'_i]$ which are spatially close to the seed s and are classified into $\widetilde{C}^{[}_{(i-1)\Omega+j}s]$ for each j ranging between 1 and Ω. The cluster expansion process is repeated by considering each sensor point already grouped in C'_k as expansion seed. Once no more sensor point can be added to C'_k, P'_k is built by sequencing the trend prototypes $\widetilde{P}^{[s]}_{(i-1)\Omega+1}$, $\widetilde{P}^{[s]}_{(i-1)\Omega+2}$, ..., and $\widetilde{P}^{[s]}_{(i-1)\Omega+\Omega}$ (line 14 in Main routine of Algorithm 1). Finally, the tend cluster (W'_i, C'_k, P'_k) is added to the set TC'_i (line 15 in Main routine of Algorithm 1).

Space-Time DRILL-DOWN. The drill-down request is formulated as follows:

`DRILL-DOWN on` M `OF GEOTRENDCUBE` $\mathcal{C}$

and triggers the drill-down process to build a new cube $\mathcal{C}'$ from $\mathcal{C}$. The cube $\mathcal{C}'$ has the same base stream of $\mathcal{C}$. The time hierarchy H'_T collects time points of T grouped, at the window level, into the windows of the 1-sized segmentation of T. For each window $W' \in window_1(T)$, the space hierarchy $H'_{space[W']}$ collects spatial points of $space[W']$ grouped, at the cluster level, into clusters containing a single point. This means that both the time point level and the window level of the hierarchy H'_T exhibit the same values. Similarly, both the spatial point level and the cluster level of each hierarchy $H'_{space[W']}$ contain the same values.

The drill-down process is iterative. At each iteration i, it pours the i-th window slice $\mathcal{C}[W_i]$ of $\mathcal{C}$ into a buffer synopsis. Let $TC_i = \{(W_i, C_k, P_k)\}_k$ be the trend cluster set which defines $\mathcal{C}[W_i]$. TC_i is processed to determine the series of trend cluster sets $\langle TC'_{(i-1)+1}, TC'_{(i-1)+2}, \ldots, TC'_{(i-1)+w}\rangle$ having window size 1. Each trend cluster set $TC'_{(i-1)+j}$ defines the window slice $\mathcal{C}'[W_{(i-1)w+j}]$ into $\mathcal{C}'$ for j ranging between 1 and w

The construction of each trend cluster set $TC'_{(i-1)+j}$ proceeds as reported in Algorithm 2. Let $t_{(i-1)w+j}$ be a time point of W_i (with j ranging between 1 and w) (line 1 in Algorithm 2). For each sensor point $s \in space[W_i]$ (line 3 in Algorithm 2), the trend cluster $(W_i, \widetilde{C}^{[s]}, \widetilde{P}^{[s]}) \in TC_i$ which contains s (i.e. $s \in \widetilde{C}^{[s]}$) is identified and its trend prototype $\widetilde{P}^{[s]}$ is selected (line 4 in Algorithm 2). The trend point $trendPoint(\widetilde{P}^{[s]}, j)$ timestamped with $t_{(i-1)w+j}$ in $\widetilde{P}^{[s]}$ is recovered

Algorithm 1. ROLL-UP: $TC_{(i-1)\Omega+1}, TC_{(i-1)\Omega+2} \ldots, TC_{(i-1)\Omega+\Omega}, \Delta \mapsto TC'_i$

– *Main routine*

1: $TC'_i \leftarrow \oslash$
Roll-Up in SPACE
2: **for all** $j = 1$ TO Ω **do**
3: $\widetilde{TC}_{(i-1)\Omega+j} \leftarrow$SUMATRA$(TC_{(i-1)\Omega+j}, w, \Delta)$
4: **end for**
Roll-up in TIME
5: $W'_i \leftarrow \bigcup_{j=1}^{w} W_{(i-1)\Omega+j}$
6: $k \leftarrow 1$;
7: **for all** ($s \in space[W'_i]$ and s is unclustered) **do**
8: **for all** ($j = 1$ TO Ω) **do**
9: $\widetilde{C}^{[s]}_{(i-1)\Omega+j} \leftarrow$ cluster$(s, \widetilde{TC}_{(i-1)\Omega+j})$
10: $\widetilde{P}^{[s]}_{(i-1)\Omega+j} \leftarrow$ trendPrototype$(s, \widetilde{TC}_{(i-1)\Omega+j})$
11: **end for**
12: $C'_k = \{s\}$;
13: $C'_k \leftarrow$ expandCluster$(s, C'_k, \{\widetilde{C}^{[s]}_{(i-1)\Omega+j}\}_{j=1,\ldots,\Omega})$
14: $P'_k = \widetilde{P}_{(i-1)\Omega+1} \bullet \widetilde{P'}_i^{(i-1)\Omega+2} \bullet \ldots \bullet \widetilde{P'}_{(i-1)\Omega+\Omega}$;
15: $TC'_i = TC'_i \cup (W'_i, C'_k, P'_k)$;
16: **end for**

– *expandCluster*$(s, C'_k, \{\widetilde{C}^{[s]}_{(i-1)\Omega+j}\}_{j=1,\ldots,\Omega}) \mapsto C'_k$

1: **for all** ($t \in space[W'_i]$ with t spatially close to s and t unclustered) **do**
2: **if** cluster$(t, \{\widetilde{C}^{[s]}_{(i-1)\Omega+j}\}_{j=1,\ldots,\Omega})$ **then**
3: $C'_k \leftarrow$ expandCluster$(\ t, C'_k \cup \{t\}, \{\widetilde{C}^{[s]}_{(i-1)\Omega+j}\}_{j=1,\ldots,\Omega}\ )$;
4: **end if**
5: **end for**

and used to define the trend cluster $([t_{(i-1)w+j}], \{s\}, \langle trendPoint(\widetilde{P}^{[s]}, j)\rangle)$ which is added to the set $TC'_{(i-1)w+j}$ (line 5 of Algorithm 2).

4 A Case Study

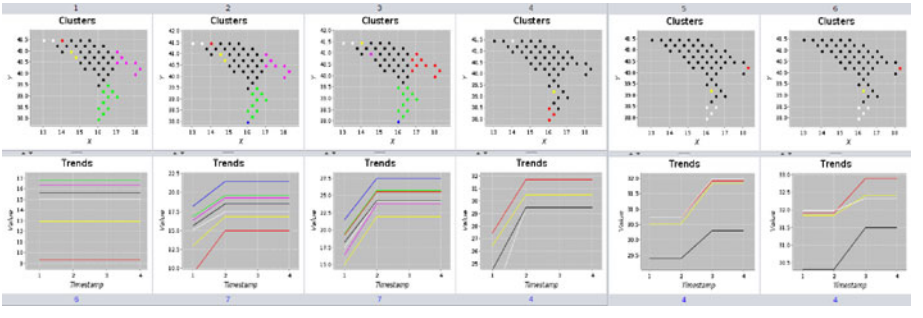
We describe an application of GeoTube to maintain the electrical power (in kw/h) weekly transmitted from PhotoVoltaic (PV) plants. The stream is generated with PVGIS[1](http://re.jrc.ec.europa. eu/pvgis) by distributing 52 PV plants over the South of Italy: each plant 0.5 degree of latitude/longitude distance apart the others. The weekly estimates of electricity production is obtained by the default parameter setting in PVGIS and streamed for the time of 52 weeks.

GeoTube is first employed to feed the geo-trend stream cube $\mathcal{C}$ constructed with $w = 4$ and $\delta = 1.5Kw/h$. The spatial closeness relation between plants is defined on the distance, i.e. two PV plants are spatially close if their distance

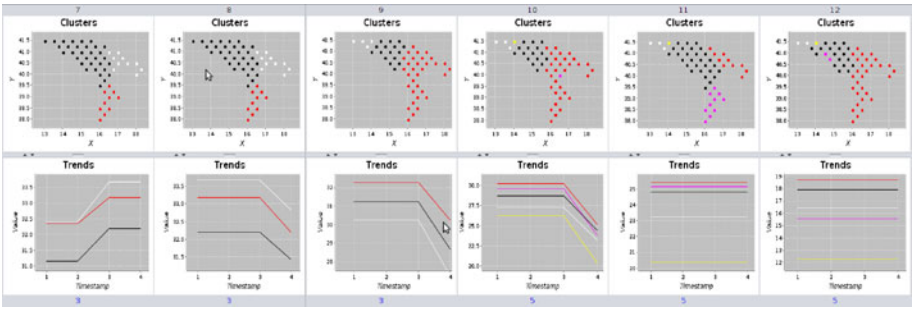
[1] PVGIS is a map-based inventory of the PV plants electricity productions.

Algorithm 2. DRILL-DOWN: $TC_i \mapsto \langle TC'_{(i-1)w+1}, TC'_{(i-1)w+2}, \ldots TC'_{(i-1)w+w} \rangle$

1: **for all** $(j = 1$ TO $w)$ **do**
2: $\quad TC'_{(i-1)w+j} = \oslash$
3: $\quad$ **for all** $(s \in space[W_i])$ **do**
4: $\quad\quad \widetilde{P}^{[s]} \leftarrow$ trendPrototype(s, TC_i)
5: $\quad\quad TC'_{(i-1)w+j} \leftarrow TC'_{(i-1)w+j} \cup ([t_{(i-1)w+j}], \{s\}, trendPoint(\widetilde{P}^{[s]}, j))$
6: $\quad$ **end for**
7: **end for**



(a) Windows 1-6



(b) Windows 7-12

Fig. 2. Geo-Trend Stream Cube construction with $w = 4$-$\delta = 1.5Kw/h$

thought the Earth is at worst 0.5 degree of latitude/longitude. Clusters and trends of windows slices of $\mathcal{C}$ are plotted in Figure 2. Then, GeoTube permits to navigate $\mathcal{C}$ one level up by rolling-up in time with $\Omega = 3$ and in space with $\Delta = 1.5$. This way we view the stream aggregated with longer window and larger clusters although past snapshots have been definitely discarded. The output cube $\mathcal{C}'$ plotted in Figure 3 permits to visualize longer trends (12 weeks) shared by larger clusters. We observe that by setting $\Omega = 1$ and $\Delta > 0$, the roll-up is only in space. Differently, by setting $\Omega > 1$ and $\Delta = 0$ the roll-up is only in time.

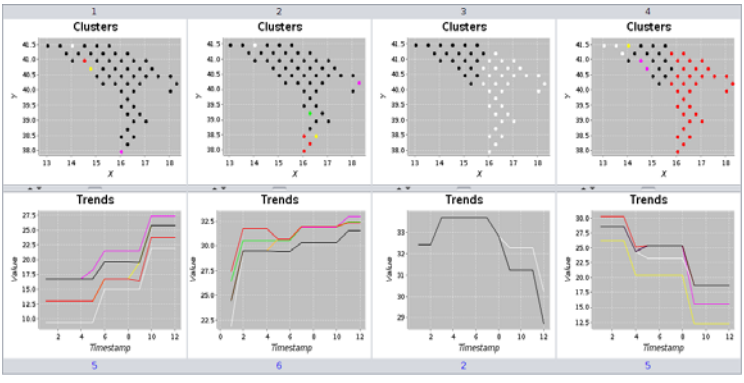


Fig. 3. Space-Time ROLL-UP with $\Delta = 1.5$ and $\Omega = 3$

Finally, we use GeoTube to navigate $\mathcal{C}$ one level down by drilling-down in both time and space. We do not plot the output cube due to space limitation, but we emphasize here that both the root mean square error and the mean absolute error are always under 0.5 ($< \delta$) if we consider accuracy of DRILL-DOWN($\mathcal{C}$) in approximating D at each time point and at each sensor point.

5 Conclusions

We define the concept of a geo-trend stream cube fed by a spatially distributed data stream. The space-time roll-up and drill-down operations are specified to navigate the cube at multiple levels of space/time granularity. These operations are supported by the system GeoTube and tested into a case study.

Acknowledgments

This work is supported by the project ATENEO 2010 entitled "Modelli e Metodi Computazionali per la Scoperta di Conoscenza in Dati Spazio-Temporali".

References

1. Ciampi, A., Appice, A., Malerba, D.: Summarization for geographically distributed data streams. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6278, pp. 339–348. Springer, Heidelberg (2010)
2. Cuzzocrea, A.: Cams: Olaping multidimensional data streams efficiently. In: Pedersen, T.B., Mohania, M.K., Tjoa, A.M. (eds.) DaWaK 2009. LNCS, vol. 5691, pp. 48–62. Springer, Heidelberg (2009)
3. Ferdous, S., Fegaras, L., Makedon, F.: Applying data warehousing technique in pervasive assistive environment. In: PETRA 2010, pp. 33:1–33:7. ACM, New York (2010)

4. Gama, J.: Knowledge Discovery from Data Streams. CRC Press, Boca Raton (2010)
5. Han, J., Chen, Y., Dong, G., Pei, J., Wah, B.W., Wang, J., Cai, Y.D.: Stream cube: An architecture for multi-dimensional analysis of data streams. Distributed Parallel Databases 18, 173–197 (2005)
6. Lee, L.H., Wong, M.H.: Aggregate sum retrieval in sensor network by distributed prefix sum data cube. In: AINA 2005, pp. 331–336. IEEE, Los Alamitos (2005)
7. Woo, H.J., Shin, S.J., Yang, W.S., Lee, W.S.: Ds-cuber: an integrated olap environment for data streams. In: CIKM 2009, pp. 2067–2068. ACM, New York (2009)

Data Access Paths in Processing of Sets of Frequent Itemset Queries

Piotr Jedrzejczak and Marek Wojciechowski

Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
Marek.Wojciechowski@cs.put.poznan.pl

Abstract. Frequent itemset mining can be regarded as advanced database querying where a user specifies the dataset to be mined and constraints to be satisfied by the discovered itemsets. One of the research directions influenced by the above observation is the processing of sets of frequent itemset queries operating on overlapping datasets. Several methods of solving this problem have been proposed, all of them assuming selective access to the partitions of data determined by the overlapping of queries, and tested so far only on flat files. In this paper we theoretically and experimentally analyze the influence of data access paths available in database systems on the methods of frequent itemset query set processing, which is crucial from the point of view of their possible applications.

1 Introduction

Frequent itemset mining [1] is one of the fundamental data mining techniques, used both on its own and as the first step of association rules generation. The problem of frequent itemset and association rule mining was initially formulated in the context of market-basket analysis, aiming at the discovery of items frequently co-occurring in customer transactions, but it quickly found numerous applications in various domains, such as medicine, telecommunications and World Wide Web.

Frequent itemset mining can be regarded as advanced database querying where a user specifies the source dataset, the mininum support threshold and (optionally) the pattern constraints within a given constraint model [7]. Frequent itemset queries are therefore a special case of data mining queries.

Many frequent itemset mining alogrithms have been developed. The two most prominent classes of algorithms are determined by the strategy of the pattern search space traversal. Level-wise algorithms, represented by the classic *Apriori* algorithm [3], follow the breadth-first strategy, whereas pattern-growth methods, among which *FP-growth* [6] is the best known, perform the depth-first search.

Although many algoritms have been proposed, effective knowledge discovery in large volumes of data remains a complicated task and requires considerable time investment. Long data mining query execution times often result in queries being collected and processed in a batch when the system load is lower. Since those queries may have certain similarities, e.g. refer to the same data, processing

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 376–385, 2011.

them concurrently rather than sequentially gives the opportunity to execute the whole set of queries much more effectively [10].

As far as processing batches of frequent itemset queries is concerned, several methods exploiting the overlapping of queries' source datasets have been developed: *Mine Merge*, independent of the frequent itemset mining algorithm used [10]; *Common Counting* [10] and *Common Candidate Tree* [5] designed for *Apriori*; *Common Building* and *Common FP-tree* [11] based on *FP-growth*. Each of these methods, in addition to the theoretical analysis, has been tested in practice with the use of flat files and direct access paths to the source dataset's partitions. In reality, however, the mined data is often stored in databases, where, depending on the selection conditions, many different access paths may be available.

The aim of this paper is both the theoretical and practical analysis of the aforementioned concurrent frequent mining methods in the light of different data access paths. As the *FP-growth* methods are adaptions of the methods developed for *Apriori*, the analysis will be conducted for *Apriori* only. *Apriori* it is the most widely implemented frequent itemset mining algorithm and the multiple source data reads it performs should make the differences between the access paths and their impact on the total execution time more noticeable.

The topics discussed in this paper can be regarded as multiple-query optimization, which was previously extensively studied in the context of database systems [9] geared towards building a global execution plan that exploits the similarities between queries. In the field of data mining, except the problem discussed in this paper, multiple-query optimization was considered in a vastly different problem of frequent itemset mining in multiple datasets [8]. Solutions similar to the ones in this paper, however, can be found in the related domain of logic programming, where a method similar to *Common Counting* has been proposed [4].

2 Multiple-Query Optimization for Frequent Itemset Queries

2.1 Basic Definitions and Problem Statement

Itemset. Let $I = \{i_1, i_2, ..., i_n\}$ be a set of literals called *items*. An *itemset* X is a set of items from I, ie. $X \subseteq I$. The *size* of the itemset X is the number of items in it.

Transaction. Let D be a *database* of transactions, where *transaction* T is a set of elements such that $T \subseteq I$ and $T \neq \emptyset$. A transaction T *supports* the item $x \in I$ if $x \in T$. A transaction T *supports* the itemset $X \subseteq I$ if it supports all items $x \in X$, ie. $X \subseteq T$.

Support. The *support* of the itemset X in the database D is the number of transactions $T \in D$ that support X.

Frequent itemset. An itemset $X \subseteq I$ is *frequent* in D if its support is no less than a given *minimum support* threshold.

Frequent itemset query. A *frequent itemset query* is a tuple $dmq = (R, a, \Sigma, \Phi, minsup)$, where R is a database relation, a is a set-valued attribute of R, Σ is a condition involving the attributes of R called *data selection predicate*, Φ is a condition involving discovered itemsets called *pattern constraint*, and *minsup* is the minimum support threshold. The result of *dmq* is a set of itemsets discovered in $\pi_a \sigma_\Sigma R$, satisfying Φ, and having support $\geq minsup$ (π and σ denote relational projection and selection operations respectively).

Elementary data selection predicates. The *set of elementary data selection predicates* for a set of frequent itemset queries $DMQ = \{dmq_1, dmq_2, ..., dmq_n\}$ is the smallest set $S = \{s_1, s_2, ..., s_k\}$ of data selection predicates over the relation R such that for each u, v ($u \neq v$) we have $\sigma_{s_u} R \cap \sigma_{s_v} R = \emptyset$ and for each dmq_i there exist integers $a, b, ..., m$ such that $\sigma_{\Sigma_i} R = \sigma_{s_a} R \cup \sigma_{s_b} R \cup .. \cup \sigma_{s_m} R$. The set of elementary data selection predicates represents the partitioning of the database determined by overlapping of queries' datasets.

Problem. Given a set of frequent itemset queries $DMQ = \{dmq_1, dmq_2, ..., dmq_n\}$, the problem of *multiple-query optimization* of DMQ consists in generating an algorithm to execute DMQ that minimizes the overall processing time.

2.2 Apriori

Introduced in [3] and based on the observation that every subset of a frequent itemset is also frequent, the *Apriori* algorithm iteratively discovers frequent itemsets of increasing size. Frequent 1-itemsets are discovered by simply counting the occurences of each item in the database. Following iterations consist of two phases: the generation phase, during which frequent itemsets from previous iteration are used to generate candidate itemsets of size 1 more, and a verification phase, during which the algorithm counts the occurences of those itemsets in the database and discards the ones that do not meet the minimum support threshold. This process is repeated until no more frequent itemsets are discovered. To avoid performing a costly inclusion test for every candidate and every read transaction, generated candidate itemsets are stored in a hash tree.

3 Review of Existing Methods

3.1 Sequential Execution

The simplest way of processing a set of frequent itemset queries is to process them sequentially using a standard algorithm like the aforementioned *Apriori*. This represents the naive approach and even though it's not an effective solution to the problem, it provides a natural reference point when evaluating other methods.

3.2 Common Counting

The *Common Counting* [10] method reduces the amount of required data reads by integrating the scans of those parts of the database that are shared by more

than one query. All queries from the set are executed concurrently in a two-phase iterative process similar to *Apriori*. Candidate generation is performed seperately for each query, with the generated candidates stored in separate hash trees. Verification, however, is performed simultaneously for all queries during a single database scan. Each database partition is therefore read only once per iteration, effectively reducing the number of I/O operations.

3.3 Common Candidate Tree

While *Common Counting* optimizes only the database reads, *Common Candidate Tree* [5] goes a step further and shares the data structures between the concurrently processed queries as well. Like in *Common Counting*, each partition is read only once per iteration, but this time a single hash tree is shared by all queries from the set, reducing the cost associated with inclusion tests. While the structure of the hash tree itself remains identical to the one used in the original *Apriori*, the candidate itemsets are modified to include a vector of counters (one for each query) and a vector of boolean flags (to track which queries generated the itemset). Candidate generation is performed seperately for each query as in *Common Counting*, with the generated sets of candidates being merged into the extended representation and put in the common hash tree afterwards. Only that single tree is then used during the verification phase, with only the appropriate counters (ie. those corresponding to queries that both generated the candidate and refer to the currently processed partition) being incremented.

3.4 Mine Merge

The *Mine Merge* [10] algorithm presents an entirely different approach. It employs the property that in a database divided into partitions, an itemset frequent in the whole database is also frequent in at least one of the partitions. *Mine Merge* first generates intermediate queries, each of them based on a single elementary data selection predicate (ie. each referring to a single database partition). Intermediate queries are then executed sequentially (for example using *Apriori*) and their results are used to create a global list of candidate itemsets[1] for each original query. A single database scan is then performed to calculate the support of every candidate itemset and discard the ones below the desired threshold, thus producing the actual results for each of the original queries.

4 Data Access Paths in Frequent Itemset Query Set Processing

4.1 Data Structures and Access Paths

In today's world the vast majority of data, including the data targeted by frequent itemset mining, is stored in relational database systems. Contemporary

[1] Such a list consists of frequent itemsets from all partitions the query refers to.

relational database management systems (DBMSs) follow the SQL standard and offer similar fundamental functionality in the sense that they store data in tables, which can be accompanied by indexes to speed-up the selection of that data. Thus, as far as the analyzed frequent itemset query processing methods are concerned we can generally assume that: (1) the data to be mined is stored in a database table, (2) each data selection predicate selects a subset of rows from that table, (3) there are two methods of accessing the rows satisfying a given data selection predicate: a full scan of the table during which the predicate is evaluated for each row, and selective access with the help of index structures.

While keeping the analysis as general and product-independent as possible, it should be noted that DBMSs available on the market compete with each other and therefore provide different choices for table organization and indexing. In the experiments accompanying our theoretical study we use Oracle 11g, considered the industry-leading database management system. Oracle 11g is an example of an object-relational DBMS, i.e. it offers object extensions to the relational model such as user-defined types and collections. We use a VARRAY collection type to store itemsets. The default table organization in Oracle 11g is heap (which is unordered). Two types of indexes are available: B-tree and bitmap indexes. We use B-trees as they support range selection predicates. An interesting alternative to a heap-organized table accompanied by an index in Oracle 11g is an index-organized table. We also consider it in the experiments.

4.2 Implementation of Compared Methods

All the compared methods were formulated in terms of reading partitions corresponding to elementary data selection predicates. Therefore, if all partitions of the table can be selectively accessed thanks to an index, all the considered methods are directly applicable with no need for extra optimizations. The question is whether these methods can avoid performing a separate full scan of the table for each partition if no applicable index is available or the query optimizer decides not to use it[2].

As for *Mine Merge*, we currently do not see any satisfactory solutions that would prevent it from suffering a significant performance loss when full scans are necessary[3]. However, it should be noted that since *Mine Merge* executes its intermediate queries independently of each other, each query can employ a different access path (a full scan or an index scan depending on the index availability and the estimated cost).

Contrary to *Mine Merge*, *Common Counting* and *Common Candidate Tree* can be implemented in a way that minimizes the negative effects of full scans.

[2] The optimizer might not use an index if a full scan results in a lower estimated cost due to poor selectivity of the index for a given selection predicate. In our discussion it is not relevant what the reason for performing a full scan to access a partition actually was.

[3] One possible solution is to materialize partitions corresponding to intermediate queries in one full table scan, but we consider it impractical for large datasets.

Both methods read multiple partitions (the ones referred to by the queries still being executed) per iteration. However, since their candidate counting is in fact performed per transaction, not per partition (individual transactions passed through hash trees), the actual order in which transactions are retrieved from the database is irrelevant. Thus, *Common Counting* and *Common Candidate Tree* can perform a single SQL query in each iteration, reading the sum of the partitions required by the queries whose execution still did not finish. This modification is crucial if full scans would be required to retrieve any individual partition (one full scan instead of several full scans and/or table accesses by index per iteration[4]) but can also be beneficial if all the partitions are accessible by index (in certain circumstances reading all the partitions in one full table scan may be more efficient than reading them one by one using an index[5]).

4.3 Theoretical Cost Analysis

In order to analyze the impact of data access paths we will provide cost formulas for the amount of data read by the compared methods for both selective access and full table scans. We will not include the cost of in-memory computations in the formulas as it does not depend on the chosen data access path. For the sake of simplicity we will assume that all *Apriori* executions (for the original as well as *Mine Merge* intermediate queries) require the same number of iterations. The variables appearing in the formulas are as follows: k - the number of *Apriori* iterations for each query, n - the number of original queries, ni - the number of *Mine Merge* intermediate queries, DB - the size of the database table containing input data, SUM - the sum of the sizes of the original queries' datasets, CVR - the total size of the parts of the table referred to (covered) by the queries.

The cost formulas for sequential execution for selective access (SEQ_{IDX}) and full table scans (SEQ_{FULL}) are presented below. For selective data access each query reads its source dataset k times. With full scans each query reads the whole table k times.

$$SEQ_{IDX} = k * SUM, \qquad SEQ_{FULL} = n * k * DB \tag{1}$$

The formulas for *Mine Merge* include the cost of the (additional) verifying scan of data. Full scan formula involves the number of intermediate queries, which is not present in the formula for selective data reads – in that case, only the amount of covered data is important, not the number of partitions into which it is divided.

$$MM_{IDX} = (k + 1) * CVR, \qquad MM_{FULL} = (ni * k + 1) * DB \tag{2}$$

Common Counting and *Common Candidate Tree* differ only in in-memory data structures, therefore the two methods share the formulas for data access

[4] Even if for just one partition a full scan is the only or the best option, all the partitions are to be retrieved in one full table scan. This is different from Mine Merge where each partition could be retrieved using a different access path.

[5] The choice of an access path is up to the query optimizer.

costs. Thanks to the integrated full scan proposed in Sect. 4.2, the cost for full scans does not depend on the number of queries (similarly as in the case of selective access).

$$CC_{IDX} = k * CVR, \qquad CC_{FULL} = k * DB \tag{3}$$

When comparing the above cost formulas one should take into account that: $n * DB \geq SUM \geq CVR$, $DB \geq CVR$, $ni \geq n$ (regarding the latter, the upper limit on ni is $2^n - 1$)[6].

Comparing the data access costs per algorithm, the increase of the cost when selective access is replaced with full scans varies among the methods: it is the smallest for *Common Counting* (independent of the number of queries) and the biggest for *Mine Merge* (dependent on the number of intermediate queries). The consequence of the above difference is a questionable applicability of *Mine Merge* if the data has to be retrieved using full table scans. With selective access *Mine Merge* should outperform sequential execution, provided the overlapping among the queries (exploited in each *Apriori* iteration) compensates for the extra scan of data[7]. With full scans *Mine Merge* can be expected to always perform worse than sequential execution. On the other hand, *Common Counting* (and *Common Candidate Tree*) not only should outperform sequential execution regardless of the available access path but even relatively benefit from full scans.

5 Experimental Results

Experiments were conducted on a synthetic dataset generated with GEN [2] using the following settings: number of transactions = 10 000 000, average number of items in a transaction = 8, number of different items = 1 000, number of patters = 15 000, average pattern length = 4. Data was stored as `<transaction id, varray of item>` pairs inside Oracle 11g database deployed on SuSE Linux, with the test application written in Java running on Mac OS X 10.6.6. Database connection was handled through JDBC over 1 Gigabit Ethernet.

Two experiments were conducted: the first one included two fixed-size queries with varying level of overlapping between them; in the second the overall scope of the processed part of the dataset was fixed while the number of fixed-size queries in the set varied. Both experiments measured the execution times of sequential execution (SEQ), *Common Counting* (CC), *Common Candidate Tree* (CCT) and *Mine Merge* (MM) for both the sequential (full scan) and selective (index scan[8]) access paths.

[6] The upper limit on the number of *Mine Merge* intermediate queries (equal to the number of elementary data selection predicates) can be smaller if certain constraints on data selection predicates are applied. For example, if all the predicates select single ranges of the same attribute, the maximal number of intermediate queries is $2 * n - 1$.

[7] In our analysis we do not consider the differences in the cost of in-memory computation, which can be a differentiator if the data access costs are identical or similar.

[8] The same experiments were repeated using an index-organized table, giving consistent results.

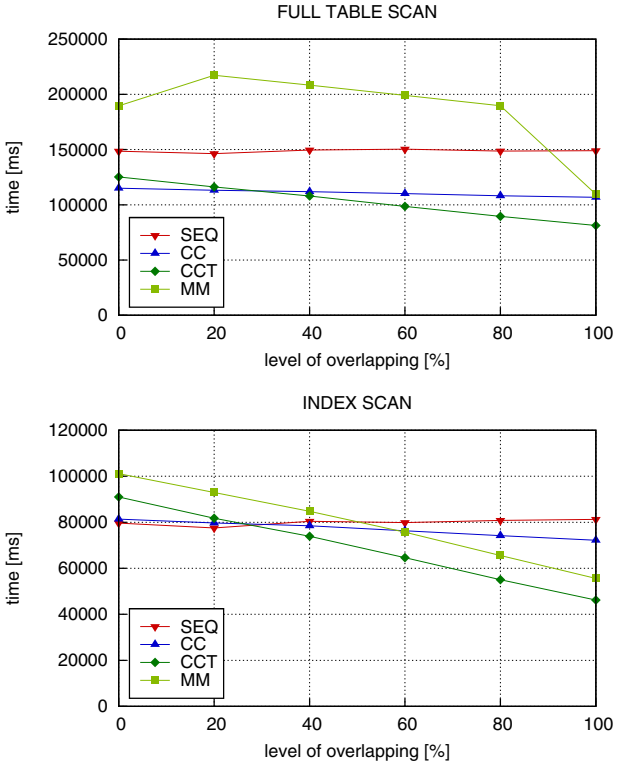


Fig. 1. Execution times for two queries and different levels of overlapping

The results of the first experiment for two queries of 1 000 000 transactions each, minimum support of 0.7%[9] and the level of overlapping from 0% to 100% are shown in Fig. 1.

As predicted, *Mine Merge* performed significantly worse than other methods without selective access, losing even with the sequential execution. With index scans available its loss wasn't as noticeable and it even managed to outperform *Common Counting* when the level of overlapping is high enough.

Both *Common Counting* and *Common Candidate Tree* performed well regardless of the access path. While their times for lower levels of overlapping were similar, *Common Candidate Tree* was clearly better when queries overlapped signficantly.

The second experiment had the queries access the same fixed part of the database each time. The query set consisted of 2 to 6 queries of size 600 000 transactions each, spread evenly across the first 1 000 000 transactions from the

[9] Experiments were conducted with two different minimum support thresholds of 0.7% and 2% with consistent results; due to limited space, only the former threshold is presented.

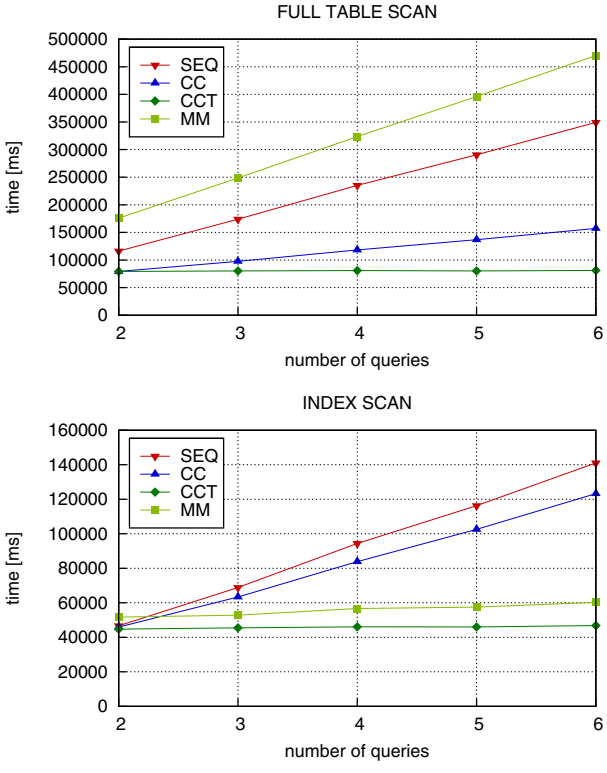


Fig. 2. Execution times for fixed scope and different numbers of queries

database (each time the first query in the set referred to transactions with identifiers from 0 to 600 000, the last one – 400 000 to 1 000 000). Results are presented in Fig. 2.

As was the case in the first experiment, *Mine Merge* was very inefficient when forced to execute full table scans, performing even worse than sequential execution. With selective access, however, the number of queries had little impact on *Mine Merge* execution times, which again allowed it to perform better than *Common Counting* and quite close to *Common Candidate Tree*, which was the fastest algorithm for both access paths. *Common Counting*, though better than sequential execution in both cases, provided a more noticeable gain over the naive method during full scans than when using the selective access path.

6 Conclusion

We considered the influence of data access paths available in DBMSs on the implementations and performance of the methods of frequent itemset query set processing designed for the *Apriori* algorithm. As expected, both the theoretical and experimental analysis showed that the performance of all the compared

methods suffers if selective access to data partitions is replaced with full scans. However, an important conclusion is that while the negative effect of full scans on *Mine Merge* is more significant than in the case of sequential processing, properly implemented *Common Counting* and *Common Candidate Tree* actually increase their advantage over sequential execution if full scans are necessary. In other words, *Mine Merge* is strongly dependent on efficient access paths to data partitions, whereas *Common Counting* and *Common Candidate Tree* can be successfully applied regardless of available data access paths.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216. ACM Press, New York (1993)
2. Agrawal, R., Mehta, M., Shafer, J.C., Srikant, R., Arning, A., Bollinger, T.: The quest data mining system. In: Simoudis, E., Han, J., Fayyad, U.M. (eds.) Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 244–249. AAAI Press, Menlo Park (1996)
3. 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 Int. Conf. on Very Large Data Bases, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
4. Blockeel, H., Dehaspe, L., Demoen, B., Janssens, G., Ramon, J., Vandecasteele, H.: Improving the efficiency of inductive logic programming through the use of query packs. Journal of Artificial Intelligence Research 16, 135–166 (2002)
5. Grudzinski, P., Wojciechowski, M.: Integration of candidate hash trees in concurrent processing of frequent itemset queries using apriori. In: Proceedings of the 3rd ADBIS Workshop on Data Mining and Knowledge Discovery, pp. 71–81 (2007)
6. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Chen, W., Naughton, J.F., Bernstein, P.A. (eds.) Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 1–12. ACM, New York (2000)
7. Imielinski, T., Mannila, H.: A database perspective on knowledge discovery. Communications of the ACM 39(11), 58–64 (1996)
8. Jin, R., Sinha, K., Agrawal, G.: Simultaneous optimization of complex mining tasks with a knowledgeable cache. In: Grossman, R., Bayardo, R.J., Bennett, K.P. (eds.) Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 600–605. ACM, New York (2005)
9. Sellis, T.K.: Multiple-query optimization. ACM Transactions on Database Systems 13(1), 23–52 (1988)
10. Wojciechowski, M., Zakrzewicz, M.: Methods for batch processing of data mining queries. In: Proceedings of the 5th International Baltic Conference on Databases and Information Systems, pp. 225–236 (2002)
11. Wojciechowski, M., Galecki, K., Gawronek, K.: Three strategies for concurrent processing of frequent itemset queries using FP-growth. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 240–258. Springer, Heidelberg (2007)

Injecting Domain Knowledge into RDBMS – Compression of Alphanumeric Data Attributes

Marcin Kowalski[1,2], Dominik Ślęzak[1,2],
Graham Toppin[2], and Arkadiusz Wojna[1,2]

[1] MIM, University of Warsaw, Poland
[2] Infobright Inc., Canada & Poland

Abstract. We discuss the framework for applying knowledge about internal structure of data values to better handle alphanumeric attributes in one of the analytic RDBMS engines. It enables to improve data storage and access with no changes at the data schema level. We present the first results obtained within the proposed framework with respect to data compression ratios, as well as data (de)compression speeds.

Keywords: Analytic RDBMS, Data Semantics, Data Compression.

1 Introduction

Data volumes that one needs to operate on daily bases, as well as the costs of data transfer and management are continually growing [6]. This is also true for analytic databases designed for advanced reports and ad hoc querying [3].

In this study, we discuss how the domain knowledge about data content may be taken into account in an RDBMS solution. By injecting such knowledge into a database engine, we expect influencing data storage and query processing. On the other hand, as already observed in our previous research [8], the method of injecting domain knowledge cannot make a given system too complicated.

As a specific case study, we consider the Infobright's analytic database engine [11,12], which implements a form of adaptive query processing and automates the task of physical database design. It also provides minimized user interface at a configuration level and low storage overhead due to data compression.

We concentrate on alphanumeric data attributes whose values have often rich semantics ignored at the database schema level. The results obtained for appropriately extended above-mentioned RDBMS engine prove that our approach can be useful for more efficient and faster (de)compression of real-life data sets.

The paper is organized as follows: Section 2 introduces the considered analytic database platform. Section 3 describes our motivation for developing the proposed framework for alphanumeric attributes. Sections 4 and 5 outline the main steps of conceptual design and implementation, respectively. Section 6 shows some experimental results. Section 7 summarizes our research in this area.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 386–395, 2011.

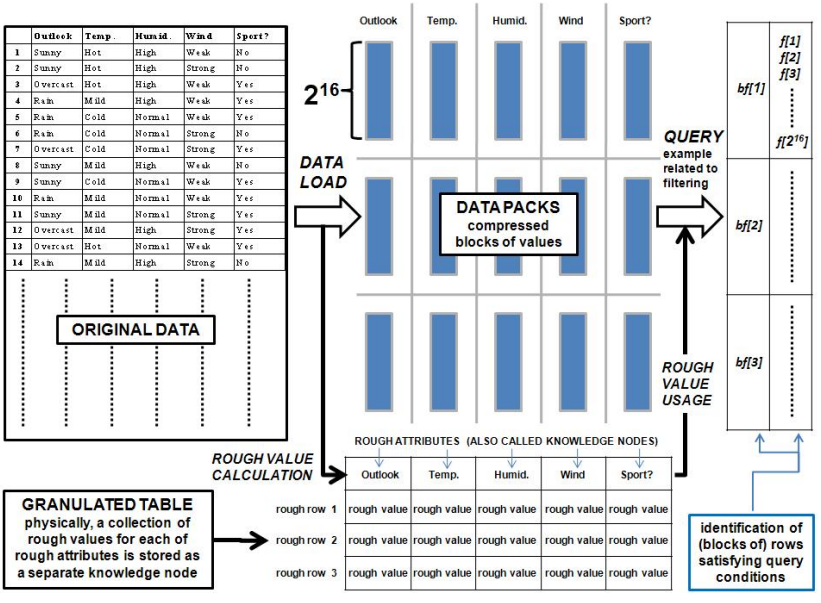


Fig. 1. Loading and querying in ICE/IEE. Rough values can be used, e.g., to exclude data packs that do not satisfy some query conditions.

2 Starting Point

In our opinion, the approach proposed in this paper might be embedded into a number of RDBMS engines that store data in a columnar way, to achieve faster data access while analytic querying [3,10]. *Infobright Community Edition* (*ICE*) and *Infobright Enterprise Edition* (*IEE*) are just two out of many possible database platforms for investigating better usage of domain knowledge about data content. On the other hand, there are some specific features of ICE/IEE architecture that seem to match with the presented ideas particularly well.

ICE/IEE creates *granulated tables* with rows (called *rough rows*) corresponding to the groups of 2^{16} original rows and columns corresponding to various forms of compact information. We refer to the layer responsible for maintaining granulated tables as to *Infobright's Knowledge Grid*.[1] Data operations involve two levels: 1) granulated tables with rough rows and their *rough values* corresponding to information about particular data attributes, and 2) the underlying repository of *data packs*, which are compressed collections of 2^{16} values of particular data attributes. Rough values and data packs are stored on disk. Rough values are small enough to keep them at least partially in memory during query sessions. A relatively small fraction of data packs is maintained in memory as well. Data packs are generally accessed on demand. We refer to [11,12] for more details.

[1] Our definition of Knowledge Grid is different than, e.g., in grid computing or semantic web [1], though the framework proposed in this paper exposes some analogies.

3 Challenges

ICE/IEE runs fast when rough values are highly *informative* and when data packs are highly compressed and easy to decompress. As already mentioned, that second aspect can be considered for some other RDBMS platforms as well. Also the first aspect might be analyzed for other databases although their usage of information analogous to rough values is relatively limited (see e.g. [5]).

By informativeness of rough values we mean that they should possibly often classify data packs as fully *irrelevant* or *relevant* at the moment of requesting those packs' status by the query execution modules. In case of full irrelevance or full relevance, the execution modules are usually able to continue with no need of decompression [12]. For the remaining *suspect* packs that require to be accessed value by value, the size of data portions to be taken from disk and the speed of making them processable by the execution modules are critical.

Achieving good quality of rough values and good (de)compression characteristics becomes harder for more complex types of attributes. Given such applications as online, mobile, or machine-generated data analytics, the issues typically arise with long varchar columns that store, e.g., URLs, emails, or texts. Moreover, even for such fields as IP or IMSI numbers that could be easily encoded as integers, the question remains at what stage of database modeling such encodings should be applied and how they may affect the end users' everyday work.

We examined many rough value structures that summarize the collections of long varchars. The criteria included high level of the above-mentioned informativeness and small size comparing to the original data packs. Some of those rough values are implemented in ICE/IEE. However, it is hard to imagine a universal structure representing well enough varchars originating from all specific kinds of applications. The same can be observed for data compression. We adopted and extended quite powerful algorithms compressing alphanumeric data [4,11]. However, such algorithms would work even better when guided by knowledge about the origin or, in other words, the internal semantics of input values.

In [8], we suggested how to use the attribute semantics while building rough values and compressing data packs. We noticed that in some applications the data providers and domain experts may express such semantics by means of the data schema changes. However, in many situations, the data schemas must remain untouched because of high deployment costs implied by any modifications. Moreover, unmodified schemas may provide the end users with conceptually simpler means for querying the data [6]. Finally, the domain experts may prefer *injecting* their knowledge independently from standard database model levels, rather than cooperating with the database architects and administrators.

An additional question is whether the domain experts are really needed to let the system know about the data content semantics, as there are a number of approaches to recognize the data structures automatically [2]. However, it is unlikely that all application specific types of value structures can be detected without a human advise. In any way, the expert knowledge should not be ignored. Thus, an interface at this level may be useful, if it is not overcomplicated.

4 Ideas

When investigating the previously-mentioned machine-generated data sets, one may quickly realize that columns declared as varchars have often heterogenous nature. Let us refer to Figure 2, precisely to a data pack related to MaybeURL column. Within the sequence of 2^{16} values, we can see *sub-collections* of NULLs, integers, strings that can be at least partially parsed along the standard URI structure,[2] as well as outliers that do not follow any kind of meaningful structure. Following [8], our idea is to deliver each of such data packs as the original string sequence to the query execution modules but, internally, store it in form of homogeneous sub-collections compressed separately. The consistency of a data pack can be secured by its *match table*, which encodes membership of each of 2^{16} values to one of sub-collections. This is an extension of our previously implemented method of decomposing data packs onto their NULL and not-NULL portions [11], applied in various forms in other RDBMS approaches as well. The difference is that here we can deal with multiple sub-types of not-NULLs.

For a given data pack, each of its corresponding sub-collections is potentially easier to compress than when trying to compress all 2^{16} values together. Each of sub-collections can be also described by separate higher-quality statistics that constitute all together the pack's rough value available to the execution modules that do not even need to be aware of its internal complexity. Data compression and rough value construction routines can further take into account that particular sub-collections gather values sharing (almost) the same structure. Going back to Figure 2, each of the values in the URI sub-collection can be decomposed onto particles such as scheme, path, authority, and so on. Sub-collections of specific particles can be compressed and summarized even better than sub-collections of not decomposed values. Again, such decompositions can be kept as transparent to the query execution modules, which refer to rough values via standardly looking functions hiding internal complexity in their implementation and, if necessary, work with data packs as sequences of *recomposed* values.

Surely, the above ideas make sense only if the domain knowledge about data content is appropriately provided to the system. According to the available literature [6] and our own experience, there is a need for interfaces enabling the data providers to inject their domain knowledge directly into a database engine, with no changes to data schemas. This way, the end users are shielded from the complexity of semantic modeling, while reaping most of its benefits. In the next section, we present one of the prototype interfaces that we decided to implement. The language proposed to express the structural complexity of attribute values is a highly simplified version of the regular expressions framework, although in future it may also evolve towards other representations [9]. The choice of representation language is actually very important regardless of whether we acquire data content information via interfaces or learn it (semi)automatically, e.g., by using some algorithms adjusting optimal levels of decomposing the original values according to hierarchical definitions recommended by domain experts.

[2] URI stands for Uniform Resource Identifier (http://www.ietf.org/rfc/rfc3986.txt).

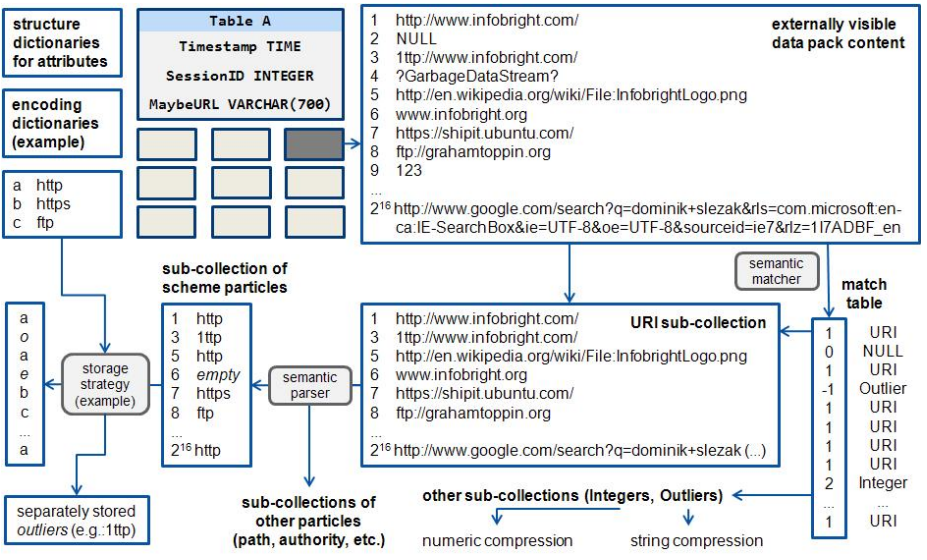


Fig. 2. Storage based on the domain knowledge. Data packs are decomposed onto sub-collections of values corresponding to different structures that can be further decomposed along particular structure specifications. It leads to sequences of more homogeneous (particles of) values that can be better compressed. For example, it is shown how to store the scheme particles of the URI values.

5 Implementation

For practical reasons, the currently implemented framework differs slightly from the previously-discussed ideas. We represent structures occurring for a given attribute (like, e.g., URIs and integers for MaybeURL, Figure 2) within a single *decomposition rule*. Such decomposition rule might be treated as disjunction of possible structures (e.g.: URI or integer or NULL or outlier), although its expressive power may go beyond simple disjunctions. The main proposed components are as follows: 1) dictionary of available decomposition rules, 2) applying decomposition rules to data attributes, and 3) parsing values through decomposition rules. These components are added to the ICE/IEE implementation integrated with MySQL framework for the *pluggable storage engines* (Figure 3).[3]

The first component – the dictionary of available decomposition rules – corresponds to the newly introduced system table *decomposition_dictionary* that holds all available decomposition rules. The table is located in the system database *sys_infobright* and is created at ICE/IEE's installation. The table contains three columns: ID (name of a decomposition rule), RULE (definition of a decomposition rule), and COMMENT (additional comments, if any). The rules can be added and modified with help of the following three stored procedures:

[3] http://dev.mysql.com/doc/refman/5.1/en/storage-engines.html

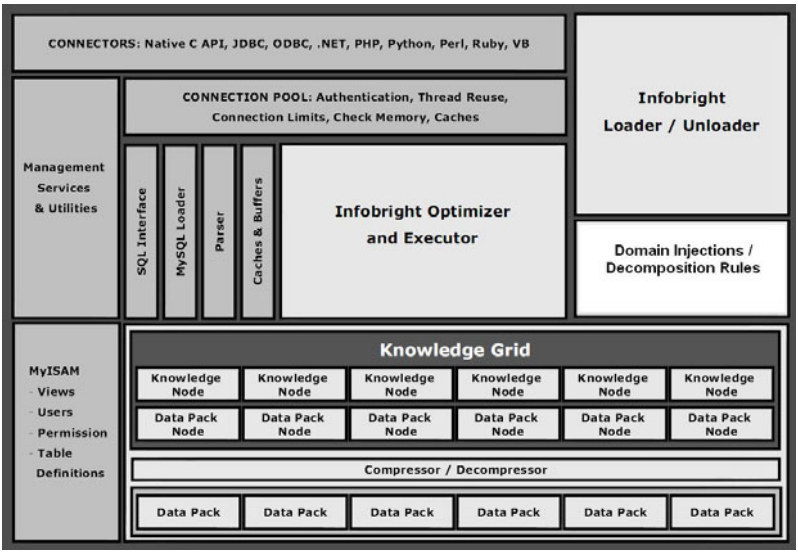


Fig. 3. Conceptual layout of the current ICE/IEE implementation (as of June 2011). The white box represents the components related to domain injections that have significant impact on the load processes and the (meta)data layers. The dark boxes are adapted from MySQL. (MySQL management services are applied to connection pooling; MyISAM engine stores catalogue information; MySQL query rewrite and parsing modules are used too.) MySQL optimization and execution pieces are replaced by the code designed to work with the compressed data packs, as well as their navigation information and rough values stored in *data pack nodes* and *knowledge nodes*, respectively. (Not to be confused with nodes known from MPP architectures [3].)

```
CREATE_RULE(id,rule,comment)
UPDATE_RULE(id,rule)
CHANGE_RULE_COMMENT(id,comment)
```

Currently, the *decomposition_ dictionary* table accepts rules defined in the simplistic language accepting concatenations of three types of primitives:

- Numeric part: Nonnegative integers, denoted as `%d`.
- Alphanumeric part: Arbitrary character sequences, denoted as `%s`.
- Literals: Sequences of characters that have to be matched exactly.

For instance, the IPv4 and email addresses can be expressed as `%d.%d.%d.%d` and `%s@%s`, respectively, where "." and "@" are literals. Obviously, this language requires further extensions, such as composition (disjunction) or Kleene closure (repeating the same pattern). Also, the new types of primitives, such as single characters, may be considered. Nevertheless, the next section reports improvements that could be obtained even within such a limited framework.

The next component – applying decomposition rules to attributes – is realized by the system table *columns* that contains four columns: DATABASE_NAME,

TABLE_NAME, COLUMN_NAME, and DECOMPOSITION. This table stores assignments of rules to data attributes identified by its first three columns. The forth column is a foreign key of ID from *decomposition_dictionary*. There are two auxiliary stored procedures provided to handle the rule assignments:

```
SET_DECOMPOSITION_RULE(database,table,column,id)
DELETE_DECOMPOSITION_RULE(database,table,column)
```

For example, the following statement

```
CALL SET_DECOMPOSITION_RULE('NETWORK','CONNECTION','IP','IPv4');
```

means that the column IP in the table CONNECTION will be handled by the decomposition rule IPv4, due to its definition in *decomposition_dictionary*.

If one of the existing rules needs to be revised by a domain expert, there are two possibilities: 1) altering the rule's definition per se if its general pattern is wrong, or 2) linking a specific data attribute to another rule. Once the rule's definition or assignment is changed, new data portions will be processed using new configuration but already existing data packs will remain unmodified.

The last component – parsing values through decomposition rules – should be considered for each data pack separately. Data packs contain in their headers information about the applied rule. Therefore, at this level, the architecture implements the above-mentioned flexibility in modifying decomposition rules – for the given attribute, different packs can be parsed using different rules.

Currently, decomposition rules affect only data storage. There are no changes at the level of rough values, i.e., they are created as if there was no domain knowledge available. Internal structure of data packs follows Figure 2. In the match table, given the above-described language limitations, there is a unique code for all values successfully parsed through the decomposition rule, with additional codes for NULLs and outliers. In future, after enriching our language with disjunctions, the codes will become less trivial and match tables will reoccur at various decomposition levels. Sub-collections to be compressed correspond to the `%d` and `%s` primitives of parsed values. A separate sub-collection contains alphanumeric outliers. At this stage, we apply our previously-developed algorithms for compressing sequences of numeric, alphanumeric, and binary values [11]. The original data packs can be reassembled by putting decompressed sub-collections together, using the match tables and decomposition rules' specifications.

As reported in the next section, the proposed framework has potential impact on data load, data size, and data access. On the other hand, it also yields some new types of design tasks. For example, the domain injections will eventually lead towards higher complexity of Infobright's Knowledge Grid, raising some interesting challenges with respect to rough values' storage and usage. One needs to remember that the significant advantage of rough values lays in their relatively small size. However, in case of long, richly structured varchar attributes, we should not expect over-simplistic rough values to be informative enough.

6 Experiments

We tested the described framework against alphanumeric columns in the real-world tables provided by the ICE/IEE users. Decomposition rules were chosen according to preliminary analysis of data samples. The rules are evaluated with respect to the three following aspects that are crucial for the users:

- **Load time**: It includes parsing input files and compressing data packs. With a decomposition rule in place, the parsing stage includes also matching the values in each of data packs against the rule's structure. For more complex rules it takes more time. On the other hand, more complex rules lead to higher number of simpler sub-collections that may be all together compressed faster than collections of original varchar values.
- **Query time**: Currently, it is related to decompression speed and to the cost of composing separately stored particles into original values. Decompression and compression speeds are not necessarily correlated. For example, for sub-collections of numeric particles our decompression routines are much faster than corresponding compression [11]. In future, query time will be reduced due to higher informativeness of rough values, yielding less frequent data pack accesses. We expect it to be more significant than any overheads related to storing and using more compound rough values.
- **Disk size**: It is primarily related to data compression ratios. The rules decompose values into particles, whose sub-collections are compressed independently by better adjusted algorithms. For example, it may happen that some parts of complex varchars are integers. Then, numeric compression may result in smaller output, even though there is an overhead related to representing packs by means of multiple sub-blocks.

Table 1 illustrates load time, query time, and disk size for the corresponding decomposition rules, measured relatively to the situation with no domain knowledge in use. We examined data tables containing single alphanumeric attributes. Results were averaged over 10 runs. Load time is likely to increase when decomposition rules are applied. However, we can also see some promising query speedups. Compression ratios are better as well, although there are counterexamples. For instance, decomposition rule `%s://%s.%s.%s/%s` did not lead to possibility of applying compression algorithms that would be adjusted significantly better to particular URI components. On the other hand, URI decomposition paid off by means of query time. In this case, shorter strings turned out to be far easier to process, which overpowered the overhead related to a need of concatenating them into original values after decompression.

Besides the data set containing web sites parsed with the above-mentioned URI decomposition rule, we considered also IPv4 addresses and some identifiers originating from the telecommunication and biogenetic applications. Such cases represent mixtures of all types of the currently implemented primitives: numerics (`%d`), strings (`%s`), and literals (such as `AA` or `gi` in Table 1).

Table 1. Experiments with three aspects of ICE/IEE efficiency: load time, query time, and disk size. Query times are reported for SELECT * FROM table INTO OUTFILE. Results are compared with domain-unaware case. For example, query time 50.1% in the first row means that the given query runs almost two times faster when the corresponding decomposition rule is used. Five data sets with single alphanumeric attributes are considered, each of them treated with at least one decomposition rule. There are five rules studied for the last set.

data type	decomposition rule	load time	query time	disk size
IPv4	`%d.%d.%d.%d`	105.8%	50.1%	105.9%
id_1	`00%d%sAA%s%d-%d-%d`	156.4%	96.1%	87.6%
id_2	`gi%d-%s_%s%d%s`	92.7%	61.8%	85.1%
URI	`%s://%s.%s.%s/%s`	135.3%	89.7%	152.6%
logs	notice 1	113.3%	88.1%	67.5%
	notice 2	113.2%	105.4%	97.0%
	notice 3	113.1%	82.2%	61.5%
	notices 1,3 generalized	103.6%	71.2%	40.9%
	notices 1,2,3 generalized	132.2%	100.4%	82.2%

The last case (denoted as logs) refers to the data set, where each value follows one of three, roughly equinumerous distinct structures (denoted as notices 1, 2, and 3) related to three subsystem sources. Given that the currently implemented language of domain injections does not support disjunction, our first idea was to adjust the decomposition rule to notice 1, 2, or 3. Unfortunately, fixing the rule for one of notices results in 66% of values treated as outliers. Nevertheless, Table 1 shows that for notices 1 and 3 it yields quite surprising improvements. We also investigated more general rules addressing multiple notices but not going so deeply into some of their details. (This means that some parts that could be finer decomposed are now compressed as longer substrings.) When using such a rule for notices 1 and 3, with 33% of outliers (for notice 2) and slightly courser way of compressing 66% of values (for notices 1 and 3), we obtained the best outcome with respect to load speed, query speed, and compression ratio. However, further rule generalization aiming at grasping also notice 2 led us towards losing too much with respect to values corresponding to structures 1 and 3.

The above example illustrates deficiencies of our current decomposition language. It also shows that the same columns can be assigned with different rules and that it is hard to predict their benefits without monitoring data and queries. It emphasizes the necessity of evolution of the domain knowledge and the need for adaptive methods of adjusting that knowledge to the data problems, which can evolve as well. Regardless of whether the optimal approach to understanding and conducting such evolution is manual or automatic, it requires gathering the feedback related to various database efficiency characteristics and attempting to translate it towards, e.g., the decomposition rule recommendations.

7 Conclusions

Allowing domain experts to describe the content of their data leads to substantial opportunities. In this paper, we discussed how to embed such descriptions into an RDBMS engine. Experiments with data compression show significant potential of the proposed approach. They also expose that more work shall be done with respect to human-computer interfaces and the engine internals.

One of our future research directions is to use domain knowledge not only for better data compression but also for better data representation. For the specific database solution discussed in this article, one may design domain-driven rough values summarizing sub-collections of decomposed data packs.

Another idea is to avoid accessing all sub-collections of required data packs. In future, we may try to use data pack content information to resolve operations such as filter or function computation, keeping a clear border between domain-unaware execution modules and domain-aware data processing.

References

1. Cannataro, M., Talia, D.: The Knowledge Grid. Commun. ACM 46(1), 89–93 (2003)
2. Chen, D., Cheng, X. (eds.): Pattern Recognition and String Matching. Kluwer Academic Publishers, Dordrecht (2002)
3. Hellerstein, J.M., Stonebraker, M., Hamilton, J.R.: Architecture of a Database System. Foundations and Trends in Databases 1(2), 141–259 (2007)
4. Inenaga, S., et al.: On-line Construction of Compact Directed Acyclic Word Graphs. Discrete Applied Mathematics (DAM) 146(2), 156–179 (2005)
5. Metzger, J.K., Zane, B.M., Hinshaw, F.D.: Limiting Scans of Loosely Ordered and/or Grouped Relations Using Nearly Ordered Maps. US Patent 6 973, 452 (2005)
6. Moss, L.T., Atre, S.: Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-support Applications. Addison-Wesley, London (2003)
7. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. Wiley, Chichester (2008)
8. Ślęzak, D., Toppin, G.: Injecting Domain Knowledge into a Granular Database Engine: A Position Paper. In: Proc. of CIKM, pp. 1913–1916. ACM, New York (2010)
9. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing, Pacific Grove (2000)
10. White, P.W., French, C.D.: Database System with Methodology for Storing a Database Table by Vertically Partitioning All Columns of the Table. US Patent 5, 794, 229 (1998)
11. Wojnarski, M., et al.: Method and System for Data Compression in a Relational Database. US Patent Application, 2008/0071818 A1 (2008)
12. Wróblewski, J., et al.: Method and System for Storing, Organizing and Processing Data in a Relational Database. US Patent Application, 2008/0071748 A1 (2008)

Extracting Conceptual Feature Structures from Text

Troels Andreasen, Henrik Bulskov,
Per Anker Jensen, and Tine Lassen

Roskilde University, CBIT, Universitetsvej 1, DK-4000 Roskilde
Copenhagen Business School, ISV, Dalgas Have 15, DK - 2000 Frederiksberg
{troels,bulskov}@ruc.dk,
{paj.isv,tla.isv}@cbs.dk

Abstract. This paper describes the an approach to indexing texts by their conceptual content using ontologies. Central to this approach is a two-phase extraction principle divided into a syntactic annotation phase and a semantic generation phase drawing on lexico-syntactic information and semantic role assignment provided by existing lexical resources. Meaningful chunks of text are transformed into conceptual feature structures and mapped into concepts in a generative ontology. By this approach, synonymous but linguistically quite distinct expressions are extracted and mapped to the same concept in the ontology, providing a semantic indexing which enables content-based search.

Keywords: Text mining, Ontologies, Lexical Ressources.

1 Introduction

To facilitate fast and accurate information retrieval from large volumes of text, some kind of indexing is needed. The text must be parsed, and indices of the most significant parts of the text must be stored based on what is identified during parsing.

Most commonly, approaches to indexing as applied in information retrieval systems are word-based.More profound parsing approaches involving linguistic analysis and use of background knowledge have still only reached an experimental level and form part of the vision of tools for the future. An ultimate goal driving research in this area is to exploit partial semantics in the sense envisioned in the Semantic Web [3]. Many ideas have been presented recently to approach this goal by using techniques from machine learning, information retrieval, computational linguistics, databases, and especially from information extraction. Various extraction principles applying rule matching have been proposed [16]. Special attention has been paid to lexico-syntactic rules [9], for instance as in [19] and [21]. Most of the approaches are concerned with automated rule learning motivated by practical considerations due to the fact that manual rule modeling in many cases is not a realistic approach. Also, methods driven by ontologies have been proposed. Ontologies provide conceptual background knowledge and thus serve as a frame of reference for introducing semantics. Content may be annotated or indexed by mappings to concepts connected by relations in the ontology. A special approach within semantic extraction from text is ontology learning [9], [22], [1], [15] and [17].

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 396–406, 2011.

1.1 The Approach

In order for a conceptual search system to work, documents must be indexed with conceptual representations rather than with word occurrences. Thus, some kind of translation mechanism is needed in order to get from a textual form to a conceptual representation.

Our principal goal is to be able to make an onto-grammatical analysis of text by performing an ontological analysis on a phrase-grammatical basis leading to conceptual representations. We propose to derive syntactic and semantic information from selection of existing lexical resources to form rules that can be used to identify and annotate semantically coherent text fragments. The rules are formed by combining lexical and syntactic information with semantic role information. The present paper explains our approach and demonstrates how rules based on a selection of lexical resources enable mapping from text fragments to an ontology, thereby providing a conceptually enriched indexing. We present a new approach to the idea of an onto-grammar as proposed in [2], aiming at developing rules which are flexible, e.g. by adding a variety of feature restrictions, including ontological affinities and syntactic and lexical restrictions. Our approach consists of a two-phase processing of text: Phase 1 provides syntactic annotation of chunks of text (words and phrases), and phase 2 combines the conceptual content of these chunks by using transformation schemes. The transformation schemes are formed by combining lexical and syntactic information enriched with semantic roles. We extract information from a range of existing resources (currently NOMLEX-plus and VerbNet) and transform it into these rules.

Our method is ontology-based in that the indexing consists of a set of concepts represented as feature structures associated with nodes in a lattice representing a generative ontology. The lattice is formed over an ordering ISA-relation enriched with a finite set of non-ordering relations whose relata are atomic or complex concepts. Generativity is achieved through a recursive nesting of concepts by virtue of the non-ordering concept relations. For an alternative approach also using generative ontologies, see [18]. Further, our method is rule-based in the sense that rules are constructed by extracting lexical, syntactic and semantic information from existing lexical resources. The rules are constructed in a way such that they may (partially) match the syntactic surface structure of texts. If a rule matches a text fragment, variables of a feature structure template are instantiated to a conceptual representation of the matched fragments.

An important aspect of our approach is nominalization of verbs, which makes it possible to interpret them as atomic concepts in the same way as nouns. This way, we can represent the conceptual content of sentences as complex concepts where the reified verbal conceptual content is attributed with the concepts realized by the verbal arguments. This approach has an additional important advantage in that semantically related nominal and verbal forms recieve identical conceptual interpretations. For example, in our approach the different linguistic forms '*The funny circus clown amused the young audience*' and '*the amusement of the young audience over the funny circus clown*' become conceptually identical.

The structure of the remainder of this paper is as follows: Section 2 describes the notions of generative ontology and conceptual feature structure, section 3 describes our approach to ontology-based identification and mapping of concepts into a generative

ontology and section 4 describes how information from existing lexical resources is combined into a unified resource. Finally, in section 5, we conclude.

2 Generative Ontologies and Conceptual Feature Structures

To provide indexing at the conceptual level, we need a notation for concepts and a formalism for the ontology in which they are situated. We introduce an approach building on conceptual feature structures. Conceptual feature structures are feature structures that denote concepts and form the basis for extending a taxonomic structure into a generative ontology [5]. A generative ontology is based on a skeleton ontology, i.e. a taxonomy situating a set of atomic concepts A in a multiple inheritance hierarchy. The generative ontology generalizes the hierarchy to a lattice and defines an extended (in principle infinite) set of concepts by introducing a set of semantic relations and by recursively introducing specializations of concepts through feature attachment. For instance, if CLOWN, CIRCUS and FUNNY are concepts in the skeleton ontology, and WRT, CHR, EXP and CAU are semantic relations (denoting "with respect to", "characterised by", "experiencer" and "cause", respectively), then *clown[WRT:circus]*, representing the concept denoted by *circus clown*, as well as *clown[WRT:circus, CHR:funny]*, representing the concept denoted by *funny circus clown*, are examples of conceptual feature structures representing concepts which are subsumed by CLOWN, cf. figure 1. More generally, given the set of atomic concepts A and the set of semantic relations R, the set of well-formed terms L is:

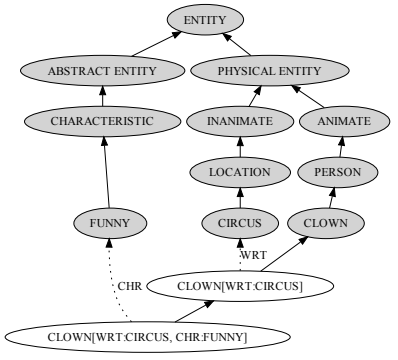


Fig. 1. Fragment of a generative ontology derived from a skeleton ontology using the semantic relations CHR and WRT. The skeleton ontology is indicated by shaded nodes; solid arrows denote subsumption, dotted arrows denote semantic relations.

$$L = \{A\} \cup \{c[r_1 : c_1, \ldots, r_n : c_n] | c \in A, r_i \in R, c_i \in L\}$$

Thus, compound concepts can have multiple and/or nested attributions. Given the skeleton ontology and the fact that any attribution gives rise to conceptual specialization, the ISA-relation defines a partial ordering connecting all possible concepts in L.

3 Ontology-Based Concept Extraction

In our approach text is processed in two phases. Phase 1 annotates the input text by lexical and syntactic tools. Phase 2 generates conceptual descriptions from the phase 1 annotations using an onto-grammar. The division of processing into these two phases is first of all a separation of annotation and description generation, which implies a

division into a syntactic and semantic processing of the text, i.e. a grammar-based and an onto-grammar-based component.

3.1 Phase 1: Annotation

In phase 1, we perform a shallow parsing of the text using lexical and syntactic tools to identify linguistic structures and annotate them. In other words, coherent text chunks such as phrases and compound words are identified, their boundaries are marked-up and they are assigned a syntactic annotation, cf. the following example

$\langle$***the***/$_{\text{DT}}$ $\langle$***funny***/$_{\text{JJ}}\rangle/_{\text{AP}}$ $\langle\langle$***circus***/$_{\text{NN}}$ ***clown***/$_{\text{NN}}\rangle/_{\text{NN}}\rangle/_{\text{NP}}$ $\langle$***amuse***/$_{\text{VBD}}\rangle/_{\text{VG}}$ $\langle$***the***/$_{\text{DT}}$ $\langle$***young***/$_{\text{JJ}}\rangle/_{\text{AP}}$ $\langle$***audience***/$_{\text{NN}}\rangle/_{\text{NP}}\rangle/_{\text{NP}}$

The rules below, expressed using the Penn Treebank tagset, are examples of syntactic rules providing the annotation.

NN ::= NN NN	VG ::= VBD
NP ::= DT NN	NP ::= DT AP NP
AP ::= JJ	NP ::= NP IN NP
NP ::= NN	

The processing of the rules lead to nested annotations and consequently the input forms a sequence of nested sequences $S = s_1 s_2 s_3 ...$ with singletons at the innermost level. Initially, singletons correspond to words in the input, but during rewriting, sequences are reduced to singletons as they are replaced by the concepts they denote. A word w annotated with a syntactic label a appears in the input as w/a. A sequence of k elements $s_1, ..., s_k$ annotated with a, where s_i is either a word or a sequence, appears as $\langle s_1, ..., s_k\rangle/a$.

Phase 1 can introduce ambiguity since processing of text can lead to several different annotations. Furthermore, ambiguity can stem from the order of application of tools and grammars. There is not necessarily one single path through phase 1, i.e. one unique sequence of application of the lexical and syntactic tools.

3.2 Phase 2: Generation

Phase 2 rewrites the annotated input from phase 1 to provide semantic descriptions in the form of conceptual feature structures (CFSs).

Rewriting is performed through so-called transformation schemes that are applied to the input. A transformation scheme is a rule that combines a pattern P, a possibly empty set of constraints C and a template T:

$$P\ C \rightarrow T$$

The pattern P matches the surface form of the annotated text given as input, and thus combines word constants, word variables and tags. The (possibly empty) constraint C expresses restrictions on the constituents of the pattern P that have to be met for the rule to apply. The template T is an expression with unbound variables that instantiates to a conceptual feature structure by unification with the pattern. A sample rule is the following:

virus/NN X/NN $\{isa(omap(X),\text{DISEASE})\} \rightarrow omap(X)$ [CAU:VIRUS]

In the pattern appears a constant word *virus* and an open variable X both with noun annotations. The function $omap(X)$ maps X to a concept in the ontology corresponding to X. The predicate expression $isa(X, Y)$ evaluates to true if X is a specialization of Y. For readability, words in the input are written in lowercase and ontology concepts in uppercase, thus $omap(disease)$ and DISEASE refer to the same ontology concept. The constraint $\{isa(omap(X),\text{DISEASE})\}$ restricts X to the ontotype DISEASE, that is, the mapping to the ontology $omap(X)$ must be a specialization of the concept DISEASE. The template in this case forms the conceptual feature structure $omap(X)$[CAU:VIRUS]. Thus, for instance *virus tonsillitis*, when annotated in phase 1 into ***virus***/NN ***tonsillitis***/NN, will by the rule above be transformed into the concept TONSILLITIS[CAU:VIRUS].

While patterns target the surface form of the annotated input, constraints and templates go beyond the input introducing functions and predicates that relate to lexical as well as ontological properties. In the example above, the function $omap$ and the predicate isa are used. Obviously new functions and predicates can be introduced to extend the expressive power, thus not only the rules but also the functions and predicates used to express the rules can be tailored to fit specific corpora, resources and ontologies. The constraint C, when included, is a set of one or more restrictions given as individual logical expressions, where the set is interpreted as a conjunction.

The principle in the rewriting is to iteratively pair a subsequence and a matching rule and to substitute the subsequence by the template from the rule. As a general restriction, to make a subsequence subject to substitution, the subsequence must be unnested, i.e. it must be a sequence of singletons. Thereby, rewriting is performed bottom-up starting at the innermost level.

The rewriting of the annotated input into a semantic description proceeds as follows:

Input: A sequence S of possibly nested annotated sequences with annotated words at the innermost level and a set of rewriting rules $\mathbf{R} = \{R_1, ..., R_n\}$

1) Select a combination of a rule $R_i : P_i\ C_i \rightarrow T_i$ and an unnested subsequence $\langle s_1 ... s_m \rangle$/a of m annotated singletons, $m \geq 1$, such that $s_1 ... s_m$ matches P_i and complies to C_i
2) Rewrite S replacing subsequence $s_1 ... s_m$ with T_i, instantiating variables unifying with P_i
3) If any remaining subsequences of S match a P_i go to 1)
4) Extract from the rewritten S all generated CFSs ignoring attached annotations

Output: A set of CFSs describing the text annotated in S

Notice that this principle is nondeterministic and varies with the order in which matching rewriting rules are selected. It can be enclosed in a deterministic derivation either by introducing a selection priority (e.g., apply always the matching rule R_i with lowest i) or by introducing an exhaustive derivation leading to all possible derivable descriptions. Further, it should be noticed that the result of repeated rewriting in 1) to 3) will not always replace all annotated sequences in S. There is no tight coupling between phase 1 and 2, but an obvious aim is to include rewriting rules in phase 2 that match most of the annotations produced in phase 1. Step 4) takes care of partially rewritten sequences: generated CFSs are extracted and everything else ignored.

As already indicated, a rule with head $x_1/_{a_1},...,x_m/_{a_m}$ matches an annotated sequence $s_1, ..., s_m$ if for all i, the annotation in s_i is a_i. Rule notation is expanded to cope with optional elements, denoted by a succeding "?" and variable length lists as indicated by variable index boundaries.

Consider as an example the following set of simple rewriting rules introducing heuristics for assignment of semantic relations between concepts constituting phrases based on some very general principles. We stipulate, by R_1 that pre-modifiers of NP-heads in the form of APs are related to the NP-head by a characterization relation (CHR) and that pre-modifying nouns (rule R_2) as well as post-modifiers of NP-heads in the form of PPs (rule R_4) are related to the head by a with-respect-to relation (WRT).

R_1 : $(X/_{\text{DT}})?\ Y/_{\text{AP}}\ Z/_{\text{NP}} \rightarrow omap(Z)$[CHR:$omap(Y)$]
R_2 : $X/_{\text{NN}}\ Y/_{\text{NN}} \rightarrow omap(Y)$[WRT:$omap(X)$]
R_3 : $(X/_{\text{DT}})?\ Y/_{\text{NP}} \rightarrow omap(Y)$
R_4 : $X/_{\text{NP}}\ Y/_{\text{IN}}\ Z/_{\text{NP}} \rightarrow omap(X)$[WRT:$omap(Z)$]

As an example, applying the transformation principle sketched above with these rules on *the funny circus clown*, two rewritings starting from the annotated input can lead to the single conceptual feature structure:
Text: *the funny circus clown*
Input: ⟨***the***/DT ⟨***funny***/JJ⟩/AP ⟨⟨***circus***/NN ***clown***/NN⟩/NN⟩/NP⟩/NP
a) ⟨***the***/DT⟨***funny***/JJ⟩/AP⟨⟨CLOWN[WRT:CIRCUS]⟩/NN⟩/NP⟩/NP
b) ⟨CLOWN[WRT:CIRCUS,CHR:FUNNY]⟩/NP
Output: CLOWN[WRT:CIRCUS,CHR:FUNNY]

Here, rule R_2 applied on the NN-NN subsequence of the input leads to a), rule R_1 applied on the remaining subsequence of a) leads to b). The result is derived by extracting feature structures (a single one in this case) from b).

Similarly below, R_1 applied on *the funny clown* leads to a), R_3 on *the circus* to b), while R_4 combines the two subconcepts leading to c):

Text: *the funny clown in the circus*
Input: ⟨***the***/DT ⟨⟨***funny***/JJ⟩/AP ***clown***/NN⟩/NP⟩/NP ***in***/IN ⟨***the***/DT ⟨***circus***/NN⟩/NP⟩/NP
a) ⟨CLOWN[CHR:FUNNY]⟩/NP ***in***/IN ⟨***the***/DT⟨***circus***/NN⟩/NP⟩/NP
b) ⟨CLOWN[CHR:FUNNY]⟩/NP ***in***/IN ⟨CIRCUS⟩/NP
c) ⟨CLOWN[CHR:FUNNY, WRT: CIRCUS]⟩/NP
Output: CLOWN[CHR:FUNNY, WRT: CIRCUS]

Notice that the two examples above also show how paraphrases can lead to identical descriptions (order of features is insignificant).

Not in all cases will all parts of the input sequence be subject to rewriting. For instance, the rules above do not "cover" all parts of the input in the following case:

Text: *the funny circus clown amused the young audience*
Input: ⟨***the***/DT ⟨***funny***/JJ⟩/AP ⟨⟨***circus***/NN ***clown***/NN⟩/NN⟩/NP⟩/NP ⟨***amuse***/VBD⟩/VG ⟨***the***/DT ⟨***young***/JJ⟩/AP ⟨***audience***/NN⟩/NP⟩/NP
a) ⟨***the***/DT ⟨***funny***/JJ⟩/AP ⟨⟨CLOWN[WRT:CIRCUS]⟩/NN⟩/NP⟩/NP ⟨***amuse***/VBD⟩/VG ⟨***the***/DT ⟨***young***/JJ⟩/AP ⟨***audience***/NN⟩/NP⟩/NP

b) ⟨CLOWN[WRT:CIRCUS,CHR:FUNNY]⟩/NP ⟨***amuse***/VBD⟩/VG ⟨***the***/DT ⟨***young***/JJ⟩/AP ⟨***audience***/NN⟩/NP⟩/NP
c) ⟨CLOWN[WRT:CIRCUS,CHR:FUNNY]⟩/NP ⟨***amuse***/VBD⟩/VG ⟨AUDIENCE[CHR:YOUNG]⟩/NP
Output: {CLOWN[WRT:CIRCUS,CHR:FUNNY], AUDIENCE[CHR:YOUNG]}

Here a) and b) correspond to the *the funny circus clown* example above while c) is the result of applying rule R_1 again to *the young audience*. Notice that *amused* is ignored here. We will return to this in the following section.

4 A Unified Lexical Resource

The semantic relations that hold between phrases or phrase elements can to some extent be deduced from existing lexical resources like VerbNet, FrameNet, WordNet, NOMLEX-plus and The Preposition Project. The resources we use at this point are VerbNet and NOMLEX-plus. However, our approach is not restricted to using these specific resources; any lexical resource may be modified and plugged into the framework. The two chosen resources are each described briefly below in sections 4.1 and 4.2, and our unified lexical resource is briefly described in section 4.3.

4.1 VerbNet

VerbNet [11] is a domain-independent, hierarchical verb lexicon for English organized into verb classes and with mapping to WordNet . Each class is specified for a subset of 24 possible thematic roles that may be associated with verbs in the class and a set of syntactic frames which specify the the argument structure and a semantic description in the form of an event structure. The thematic role information includes selectional restrictions which constrain the semantic types of the arguments associated with a thematic role. Some frames also restrict lexical items, e.g. prepositions and particles.

4.2 NOMLEX and NOMLEX-Plus

NOMLEX is a lexicon of nominalizations and their corresponding verbs. For each verb, more than one nominalization may exist, and for each nominalization, a number of possible syntactic realization patterns are described. NOMLEX-plus is a semi-automatically produced extension of NOMLEX, that specifies approximately 60 different verb subcategorization patterns. In addition to information about the syntactic form of the verbal expression, the patterns describe the grammatical function of the verbal arguments, and how they can be realized in a nominalized expression. Other types of information concern, e.g., the type of nominalization (event or state expressions of the verb, or incorporations of a verbal argument), ontological type-restrictions on arguments, plural or singular forms and frequency information.

4.3 Unifying Lexical Resources

From the selected resources, we extract, transform and combine information into a unified lexical resource. This resource contains additional information in the form of syntactic scrambling rules since neither of the included resources specifies this type of

information. From VerbNet, we extract information about verb form, semantic roles, selectional restrictions and syntactic frames, and from NOMLEX-plus, we extract information about nominalized form, verb form, syntactic frames, and possible morphological and syntactic realizations of nominal complements.

To a certain extent, the information from the resources overlaps, but it may also be distinct for a specific resource. Since VerbNet is the only resource in our selection of resources that contains role-information, we apply a unification approach: When a syntactic frame for a lexical entry from NOMLEX-plus matches a frame from VerbNet containing the same lexical item, we unify the information. If two frames do not match completely, we retain two separate frames. If a given syntactic frame is not in VerbNet, and we thus do not have semantic role-information, we add the generic role *R* to all frame items, except for the subject NP. For subject NPs, we apply a heuristic such that for a given verb, the role specified in VerbNet for the subject NP is applied to all frames in the unified entry irrespective of the frame source.

Verb pattern rules:

– Example: *the funny circus clown amused the young audience*

R_6 : X_1/NP X_2/VG X_3/NP $\{head(X_2) = amuse, isa(omap(X_3), \text{ANIMATE})\} \rightarrow$
$omap(amusement)$[CAU:$omap(X_1)$,EXP:$omap(X_3)$]

– Example: *the audience the clown amused*

R_7 : X_1/NP X_2/NP X_3/VG $\{head(X_3) = amuse, isa(omap(X_1), \text{ANIMATE})\} \rightarrow$
$omap(amusement)$[CAU:$omap(X_2)$,EXP:$omap(X_1)$]

– Example: *the audience was amused by the clown*

R_8 : X_1/NP X_2/VG X_3/IN X_4/NP $\{head(X_2) = amuse, isa(omap(X_1), \text{ANIMATE}),$
$X_2 = by\} \rightarrow$
$omap(amusement)$[CAU:$omap(X_4)$,EXP:$omap(X_1)$]

– Example: *the audience was amused*

R_9 : X_1/NP X_2/VG $\{head(X_2) = amuse, isa(omap(X_1), \text{ANIMATE})\} \rightarrow$
$omap(amusement)$[EXP:$omap(X_1)$]

Nominalization rules:

– Example: *the amusement of the young audience over the funny circus clown*

R_{10} : X_1/NP X_2/IN X_3/NP X_4/IN X_5/NP
$\{head(X_1) = amusement, isa(omap(X_3), \text{ANIMATE}),$
$X_2 = of, X_4 \in \{at, in, over, by\}\} \rightarrow$
$omap(X_1)[CAU : omap(X_5), EXP : omap(X_3)]$

– Example: *the amusement over the funny circus clown of the young audience*

R_{11} : X_1/NP X_2/IN X_3/NP X_4/IN X_5/NP
$\{head(X_1) = amusement, isa(omap(X_5), \text{ANIMATE}),$
$X_4 = of, X_2 \in \{at, in, over, by\}\} \rightarrow$
$omap(X_1)[CAU : omap(X_3), EXP : omap(X_5)]$

– Example: *the young audience's amusement over the funny circus clown*

R_{12} : X_1/NP X_2/POS X_3/NP X_4/IN X_5/NP
$\{head(X_3) = amusement, isa(omap(X_1), \text{ANIMATE}), X_4 \in \{at, in, over, by\}\} \rightarrow$
$omap(X_3)[CAU : omap(X_5), EXP : omap(X_1)]$

– Example: *the amusement of the young audience*

R_{13} : X_1/NP X_2/IN X_3/NP $\{head(X_1) = amusement, isa(omap(X_3), \text{ANIMATE}),$
$X_2 = of\} \rightarrow omap(X_1)[EXP : omap(X_3)]$

– Example: *the young audience's amusement*

R_{14} : $X_1/_{NP}\ X_2/_{POS}\ X_3/_{NP}\ \{head(X_3) = amusement, isa(omap(X_1), \text{ANIMATE}), \} \rightarrow omap(X_3)[EXP : omap(X_1)]$

– Example: *the amusement over the funny circus clown*

R_{15} : $X_1/_{NP}\ X_2/_{IN}\ X_3/_{NP}\ \{head(X_1) = amusement, X_2 \in \{at, in, over, by\}\} \rightarrow omap(X_1)[CAU : omap(X_3)]$

– Example: *the amusement*

R_{16} : $X_1/_{NP}\ \{head(X_1) = amusement\} \rightarrow omap(X_1)$

4.4 Applying the Unified Ressource

As an example of how to utilize the extracted rules presented in section 4.3 we consider a continuation of the last example in section 3.2. The development reached the point:

Text: *the funny circus clown amused the young audience*
...
c) ⟨CLOWN[WRT:CIRCUS,CHR:FUNNY]⟩/$_{NP}$ ⟨***amuse***/$_{VBD}$⟩/$_{VG}$ ⟨AUDIENCE[CHR:YOUNG]⟩/$_{NP}$

Notice that this (NP-VG-NP) sequence matches rule R_6 above, thus we can bring the development one step further by applying this rule leading to d):

d) ⟨AMUSEMENT[CAU:CLOWN[WRT:CIRCUS, CHR:FUNNY],
EXP:AUDIENCE[CHR:YOUNG]⟩/NP
Output: {AMUSEMENT[CAU:CLOWN[WRT:CIRCUS, CHR:FUNNY],
EXP:AUDIENCE[CHR:YOUNG]}

thus R_6 is applied to join the two concepts CLOWN[WRT:CIRCUS,CHR:FUNNY] and AUDIENCE[CHR:YOUNG] based on the role information associated with the verb *amuse* given in R_6.

A last example that also utilizes information extracted from NomLex and thereby exemplifies paraphrasing by normalization is the following:

Text: *the amusement of the young audience over the funny circus clown*
Input: ⟨***the***/$_{DT}$ ***amusement***/$_{NN}$⟩/$_{NP}$ ***of***/$_{IN}$ ⟨***the***/$_{DT}$ ⟨***young***/$_{JJ}$⟩/$_{AP}$ ⟨***audience***/$_{NN}$⟩/$_{NP}$⟩/$_{NP}$ ***over***/$_{IN}$ ⟨***the***/$_{DT}$ ⟨***funny***/$_{JJ}$⟩/$_{AP}$ ⟨⟨***cirkus***/$_{NN}$ ***clown***/$_{NN}$⟩/$_{NN}$⟩/$_{NP}$⟩/$_{NP}$
b) ⟨***the***/$_{DT}$ ***amusement***/$_{NN}$⟩/$_{NP}$ ***of***/$_{IN}$ ⟨AUDIENCE[CHR:YOUNG]⟩/$_{NP}$ ***over***/$_{IN}$ ⟨***the***/$_{DT}$ ⟨***funny***/$_{JJ}$⟩/$_{AP}$ ⟨⟨***cirkus***/$_{NN}$ ***clown***/$_{NN}$⟩/$_{NN}$⟩/$_{NP}$⟩/$_{NP}$
c) ⟨***the***/$_{DT}$ ***amusement***/$_{NN}$⟩/$_{NP}$ ***of***/$_{IN}$ ⟨AUDIENCE[CHR:YOUNG]⟩/$_{NP}$ ***over***/$_{IN}$ ⟨***the***/$_{DT}$ ⟨***funny***/$_{JJ}$⟩/$_{AP}$ ⟨CLOWN[WRT:CIRCUS]⟩/$_{NP}$⟩/$_{NP}$
d) ⟨***the***/$_{DT}$ ***amusement***/$_{NN}$⟩/$_{NP}$ ***of***/$_{IN}$ ⟨AUDIENCE[CHR:YOUNG]⟩/$_{NP}$ ***over***/$_{IN}$ ⟨CLOWN[WRT:CIRCUS,CHR:FUNNY]⟩/$_{NP}$
e) ⟨AMUSEMENT[CAU:CLOWN[WRT:CIRCUS, CHR:FUNNY],
EXP:AUDIENCE[CHR:YOUNG]⟩/$_{NP}$
Output: {AMUSEMENT[CAU:CLOWN[WRT:CIRCUS,CHR:FUNNY],
EXP:AUDIENCE[CHR:YOUNG]}

Here the development to the point d) applies the basic heuristic rules R_1 – R_4 similar to previous examples in section 3, while use of R_{10} leads to a result that is identical to the result of extracting from the un-normalized version above.

5 Conclusions and Future Work

We have described an approach to indexing text by its conceptual content. The general idea is to combine the use of generative ontologies with syntactic grammars and lexico-syntactic information in the derivation of conceptual feature structures that represent the meaning of text independently of linguistic expression.

The OntoGram-approach divides text processing into an annotation and a generation phase. In the annotation phase, in principle, any pos-tagger and phrase grammar could be plugged in. The generation phase is driven by a set of rewriting rules which transform annotated text into conceptual feature structures. In this transformation phase, words are mapped to concepts in the ontology and semantic roles are identified and represented as features in the generated conceptual feature structures. A crucial property of this framework is that it provides an open platform where diverse lexical resources can be exploited, i.e. modeled as a set of rules for generating specific conceptual feature structures that reflect the knowledge of the given resource. For the current experiments, our focus has been on NOMLEX-plus and VerbNet, and our observation is that these resources can make a significant contribution to conceptual indexing. However, the framework is open for modeling aspects from in principle any linguistic resource that includes semantics, e.g. role-assignment and selectional restrictions. In addition, the framework can include heuristic rules that are not necessarily extracted from a specific resource. Large-scale experiments and evaluation have not yet been carried out, but we anticipate that the results will match the good results from the small-scale experiments carried out during the developement phase.

Problem areas for our approach include compounds and prepositions. Compounding (e.g. [20, 13]) as well as automatic deduction of semantic relations in noun-noun compounds (e.g. [6, 7, 8, 4, 10]) are areas which have received much attention in recent years. NOMLEX only concerns deverbal and relational nouns and, thus, for an improved treatment of noun compounds in general, we still need to look more into potential resources and theories.

Prepositions may denote a range of semantic relations and thus, our heuristic of assigning the WRT-role to all PP complements is very coarse. For a more detailed analysis of the relations denoted by prepositions, we propose to apply the principles described in [12] or using The Preposition Project ([14]). By applying such knowledge, we may be able to specify the relation between phrase heads and complements as PPs based on more subtle principles than the ones currently applied.

References

1. Agichtein, E., Gravano, L.: Snowball - extracting relations from large plain-text collections. In: Proceedings of the 5th ACM International Conference on Digital Libraries, pp. 85–94(2000)
2. Andreasen, T., Fischer Nilsson, J.: Grammatical specification of domain ontologies. Data & Knowledge Engineering 48(2), 221–230 (2004)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. In: Scientific American Magazine (2001)

4. Costello, F.J., Veale, T., Dunne, S.: Using wordnet to automatically deduce relations between words in noun-noun compounds. In: Proceedings of COLING/ACL (2006)
5. Fischer Nilsson, J.: A logico-algebraic framework for ontologies ontolog. In: Anker Jensen, P., Skadhauge, P. (eds.) Proceedings of the First International. OntoQuery Workshop,Ontology-based Interpretation of Noun Phrases,Kolding. Department of Business Communication and Information Science. University of Southern Denmark, Denmark (2001)
6. Girju, R., Moldovan, D., Tatu, M., Antohe, D.: On the semantics of noun compounds. Computer Speech and Language 19, 479–496 (2005)
7. Girju, R., Beamer, B., Rozovskaya, A., Fister, A., Bhat, S.: A knowledge-rich approach to identifying semantic relations between nominals. Information Processing and Management 46, 589–610 (2009a)
8. Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., Yuret, D.: Classification of semantic relations between nominals. Language Resources and Evaluation 43, 105–121 (2009b)
9. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the Fourteenth International Conference on Computational Linguistics, Nantes, France, pp. 539–545 (1992)
10. Kim, S.N., Baldwin, T.: Automatic Interpretation of Noun Compounds Using WordNet Similarity. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 945–956. Springer, Heidelberg (2005)
11. Kipper-Schuler, K.: VerbNet: A broad-coverage, comprehensive verb lexicon. Ph.D. thesis, Computer and Information Science Dept. University of Pennsylvania, Philadelphia (2006)
12. Lassen, T.: Uncovering Prepositional Senses, Computer Science Research Report vol. 131. Ph.D. thesis, Computer Science dept., CBIT, Roskilde University (2010)
13. Lieber, R., Stekauer, P. (eds.): The Oxford Handbook of Compounding. Oxford Handbooks. Oxford University Press, Oxford (2009)
14. Litkowski, K.C., Hargraves, O.: The preposition project. In: ACL-SIGSEM Workshop on The Linguistic Dimensions of Prepositions and Their Use in Computational Linguistic Formalisms and Applications, pp. 171–179 (2005)
15. Maedche, A., Staab, S.: Learning ontologies for the semantic web. In: Semantic Web Worskhop (2001)
16. Muslea, J.: Extraction patterns for information extraction tasks: A survey. In: AAAI 1999 Workshop on Machine Learning for Information Extraction (1999)
17. Velardi, P., Navigli, R.: Learning domain ontologies from document warehouses and dedicated web sites. Computational Linguistics 30(2), 151–179 (2004)
18. Fischer Nilsson, J., Szymczak, B., Jensen, P.A.: ONTOGRABBING: Extracting information from texts using generative ontologies. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 275–286. Springer, Heidelberg (2009)
19. Phillips, E., Riloff, W.: Exploiting role-identifying nouns and expressions for information extraction. In: 2007 Proceedings of RANLP 2007 (2007)
20. Scalise, S., Vogel, I. (eds.): Cross-Disciplinary Issues in Compounding. Current Issues in Linguistic Theory, vol. 311. John Benjamins, University of Bologna / University of Delaware (2010)
21. Serban, A., ten Teije, R., van Harmelen, F., Marcos, M., Polo-Conde, C.: Extraction and use of linguistic patterns for modelling medical guidelines. Elsevier Arificial Intelligence in Medicine 39(2), 137–149 (2007)
22. Sánchez, D., Moreno, A.: Pattern-based automatic taxonomy learning from the web. AI Communications Archive 21(1), 27–48 (2008)

Semantically-Guided Clustering of Text Documents via Frequent Subgraphs Discovery

Rafal A. Angryk[1], M. Shahriar Hossain[2], and Brandon Norick[1]

[1] Department of Computer Science, Montana State University, Bozeman, MT 59717, USA
[2] Department of Computer Science, Virginia Tech Blacksburg, VA 24061, USA
angryk@cs.montana.edu, msh@cs.vt.edu, brandon.norick@msu.montana.edu

Abstract. In this paper we introduce and analyze two improvements to GDClust [1], a system for document clustering based on the co-occurrence of frequent subgraphs. GDClust (Graph-Based Document Clustering) works with frequent senses derived from the constraints provided by the natural language rather than working with the co-occurrences of frequent keywords commonly used in the vector space model (VSM) of document clustering. Text documents are transformed to hierarchical document-graphs, and an efficient graph-mining technique is used to find frequent subgraphs. Discovered frequent subgraphs are then utilized to generate accurate sense-based document clusters. In this paper, we introduce two novel mechanisms called the Subgraph-Extension Generator (SEG) and the Maximum Subgraph-Extension Generator (MaxSEG) which directly utilize constraints from the natural language to reduce the number of candidates and the overhead imposed by our first implementation of GDClust.

Keywords: graph-based data mining, text clustering, clustering with semantic constraints.

1 Introduction

There has been significant increase in research on text clustering in the last decade. Most of these techniques rely on searching for identical words and counting their occurrences [3]-[5]. The major motivation for our approach comes from typical human behavior when people are given the task of organizing multiple documents. As an example, consider the behavior of a scientific book editor who is faced with the complicated problem of organizing multiple research papers into a single volume with a hierarchical table of contents. Typically, even papers from the same research area are written (1) in multiple writing styles (e.g., using "clusters" instead of "concentration points"), (2) on different levels of detail (e.g., survey papers versus works discussing a single algorithm) and (3) in reference to different aspects of an analyzed area (e.g., clustering of numeric versus categorical data). Instead of searching for identical words and counting their occurrences [3]-[5], the human brain usually remembers only a few crucial keywords and their abstract senses, which provide the

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 407–417, 2011.

editor with a compressed representation of the analyzed document. These keywords, discovered thanks to the expert's knowledge (replaced, in our case, by WordNet [6]-[7]), are then used by the book editor to fit a given research paper into a book's organization scheme, reflected by the table of contents.

In this paper we improve the GDClust [1], a system that performs frequent subgraph discovery from a text repository for document clustering. In our approach document similarity is evaluated by generating a graph representation of each document, where edges in the graph represent hypernym relationships of noun keywords and their abstract terms. Thus, a document-graph represents the structure within the ontology, which is independent of its specific keywords and their frequencies. In [1], we have shown that GDClust, which relies less on the appearance of specific keywords, is more robust than the traditional vector space model-based clustering and represents an advanced method for organizing the numerous documents available to us on a daily basis. Here, we propose a novel Subgraph-Extension Generation (SEG) mechanism that significantly outperforms a well-known FSG [2] approach, used in the original implementation of GDClust [1]. Additionally, we enhanced our SEG by introducing our Maximum Subgraph-Extension Generation (MaxSEG) mechanism which provides faster dissimilarity matrix construction with the cost of slightly deteriorated performance compared to our SEG approach.

The rest of the paper is organized as follows. Sec. 2 describes the background of this work. The overall GDClust system paired with our enhancements are portrayed in Sec. 3. Some illustrative experimental results and conclusions are presented in Sec. 4.

2 Background

Graph models have been used in complex datasets and recognized as useful by various researchers in chemical domain [9], computer vision technology [10], image and object retrieval [11] and social network analysis [12]. Nowadays, there are well-known subgraph discovery systems like FSG (Frequent Subgraph Discovery) [2], gSpan (graph-based Substructure pattern mining) [13], DSPM (Diagonally Subgraph Pattern Mining) [14], and SUBDUE [15]. Most of them have been tested on real and artificial datasets of chemical compounds.

Agrawal et al. [8] proposed the Apriori approach for frequent patterns discovery. It is an iterative algorithm, where candidates for larger frequent patterns are gradually grown from frequent patterns of a smaller size. We start from discovering frequent patterns of size k=1, and in the next iteration merge them into the candidates for k=2-size frequent patterns. After the candidates are created, we check for frequencies of their occurrences, and move on to the next iteration.

There had been extensive research work in the area of generating association rules from frequent itemsets [16–17]. There are also some transaction reduction approaches proposed by Han et al. [18]. We apply our own variation of mining multilevel association rules [18] for the frequent sense discovery process and utilize the Gaussian minimum support strategy to prune edges from multiple levels of the terms.

Our system (GDClust [1]) utilizes the power of using graphs to model a complex sense of text data. It constructs the document-graphs from text documents and a semantic lexicon, WordNet [6]-[7], and then applies an Apriori paradigm [8] to discover

frequent subgraphs from them. GDClust uses frequent subgraphs as the representation of common abstract senses among the document-graphs. The benefit of GDClust is that it is able to group documents in the same cluster even if they do not contain common keywords, but still possess the same sense.

The work we managed to find that is closest to our approach is a graph query refinement method proposed by Tomita et al. [19]. Their prototype depends on user interaction for the hierarchic organization of a text query. In contrast, we depend on a predefined ontology [6-7], for the automated retrieval of frequent subgraphs from text documents. GDClust offers an unsupervised system that utilizes an efficient subgraph discovery technique to harness the capability of sense-based document clustering.

3 System Overview

Our GDClust pipeline can be divided into three major components: (1) the Conversion Unit and (2) the Subgraph Discovery Unit and (3) the Clustering Unit.

The Conversion Unit is responsible for the conversion of each document to its corresponding graph representation. It utilizes the Word Vector Tool (WVTool) [20] for the retrieval of meaningful keywords from the documents. It uses the WordNet hypernymy hierarchy to construct the document-graphs. We utilized WordNet's noun taxonomy, which provides hypernymy-hyponymy relationships between concepts and allows the construction of a document-graph with up to 18 levels of abstractions. Our document-graph construction algorithm traverses up to the topmost level of abstractions of the keywords of a document to construct the corresponding document-graph. To incorporate natural language constraints and speed up the process of frequent subgraph discovery, we also construct the MDG, which is a merged document-graph containing connections between all the keywords of all the documents and their abstract terms. Section 3.2 describes how the MDG helps in faster generation of candidate subgraphs.

The Subgraph Discovery Unit discovers frequent subgraphs representing frequent senses from the generated document-graphs. The Clustering Unit constructs the dissimilarity matrix and clusters the documents utilizing the frequent subgraphs that were discovered by the Subgraph Discovery Unit. Sections 3.2 and 3.3 describe the subgraph discovery processes and the clustering mechanism used by our GDClust.

3.1 Candidate Subgraph Pruning Using Gaussian Minimum Support

We use an Apriori paradigm [4], to mine the frequent subgraphs from the document-graphs, and we utilize our Gaussian minimum support strategy to logically prune *1*-edge subgraphs from candidate list before generating any higher order subgraphs. Since using WordNet results in a very large graph of all English nouns, we introduced the MDG and proposed a Gaussian minimum support strategy in GDClust. We use the minimum support strategy to limit the number of candidate subgraphs with extremely abstract and very specific synsets. Since WordNet's ontology merges to a single term, the topmost level of abstraction is a common vertex for all of the generated document-graphs, yielding subgraphs that include the vertex with the topmost level of abstraction to be less informative for clustering. Moreover, terms near to the

bottom of the hierarchy are less useful due to their rare appearance in the document-graphs causing them to be of little use for clustering purposes. Terms appearing within the intermediate levels of the taxonomy seem to be more representative clusters' labels than subgraphs containing terms at higher and lower levels.

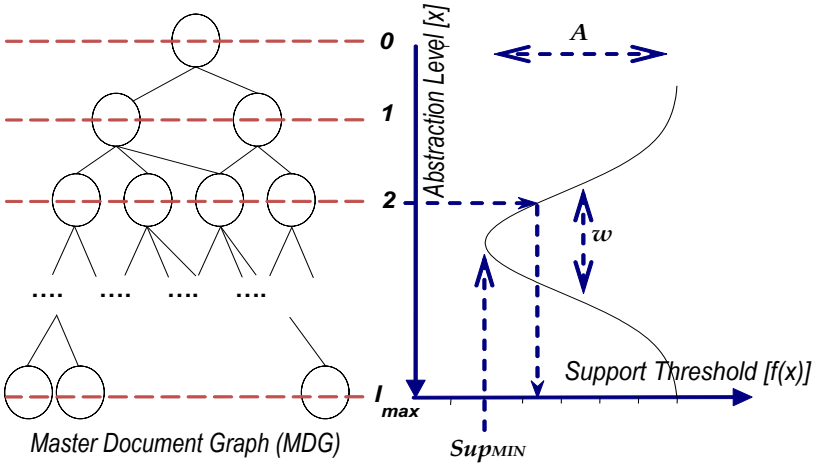


Fig. 1. Abstraction-constrained Gaussian Minimum Support

Our dynamic minimum support strategy, based on Gaussian function used to model support distribution, is illustrated in Fig. 1. The hierarchy drawn in the figure indicates our MDG, where the nodes indicate the noun keywords and their abstracts, w is the width of the Gaussian curve at $A/2$, and l_{max} is the maximum number of abstraction levels in the MDG. Since there are 18 levels of abstraction in WordNet's noun taxonomy, in our MDG $0 \leq l_{max} < 18$. Our Gaussian model accepts only the keywords which have $frequency \geq Sup_{MIN}$ (our predefined minimum support threshold) only when they appear at the mid level of our abstraction-based taxonomy. In the remaining cases, the model applies a gradual increase of minimum support threshold at higher and lower levels. This model makes the mid-levels of the taxonomy formed by MDG more important. It also assumes, based on our observation, that the generated document-graphs contain a lot of frequent, but uninteresting subgraphs at the topmost level. At the same time, the document-graphs have distinct subgraphs at the bottom levels which are not frequent enough to carry significant meaning for the clustering purposes. The first group of subgraphs would generate large clusters with low inter-cluster similarity, and the second would generate a huge number of very small clusters. We apply the Gaussian dynamic minimum support strategy to prune *1*-edge subgraphs before the starting of generation of higher order subgraphs.

3.2 Semantically-Guided Candidate Subgraph Generation

Our document-graph construction algorithm ensures that a document-graph does not contain more than one edge between two nodes. Additionally, the overall subgraph discovery concept ensures that the same subgraph does not appear more than once in

a particular document-graph. All the edges and nodes of a document-graph are uniquely labeled. We developed a fast method to generate candidate subgraphs named Subgraph-Extension Generator (SEG). We have compared our approach with the original FSG-based [2] mechanism. Additionally, we enhanced our SEG to Maximum Subgraph-Extension Generator (MaxSEG) which generates smaller number of subgraphs and thus offers faster dissimilarity matrix construction during the clustering phase. The descriptions of FSG, SEG and MaxSEG approaches are as follows.

1) *FSG* [2]: In the FSG approach, a (k+1)-edge candidate subgraph is generated by combining two k-edge subgraphs where these two k-edge subgraphs have a common core subgraph [2] of (k-1)-edges. This approach requires time-consuming comparisons between core subgraphs during the generation of a higher level subgraph. To avoid edge-by-edge comparisons, we assigned a unique code for each subgraph from the list of their edges' DFS-codes. This code is stored as the hash-code of the subgraph object. Therefore, checking two core subgraphs for equality has been reduced to a simple integer hash-code comparison. Although this approach is very fast for small graphs, it becomes inefficient for big document-graphs due to large number of blind attempts to combine *k*-edge subgraphs to generate *(k+1)*-edge subgraphs.

Consider an iteration in which we have a total of 21 *5*-edge subgraphs in the candidate list L_5. We try to generate *6*-edge subgraphs from this list. Consider the situation of generating candidates using one *5*-edge subgraph (e.g., *lmnop*) of L_5. The original FSG [2] approach tries to combine all 20 remaining subgraphs with *lmnop* but succeeds, let us assume, only in three cases. Fig. 2 illustrates that *lmnop* is successfully combined with only *mnopq*, *mnopr* and *mnops*. All 17 other attempts to generate a *6*-edge subgraph with *lmnop* fail because the *4*-edge core-subgraphs, analyzed in this case, do not match. Fig. 2 shows the attempts to generate good candidates for just one subgraph (*lmnop*). For all the subgraphs in L_5, there would be a total of $21 \times 20 = 420$ blind attempts to generate *6*-edge subgraphs. Some of these attempts would succeed, but most would fail to generate acceptable *6*-edge candidates. Although GDClust utilizes hash-codes of subgraphs and core-subgraphs for faster comparisons, it cannot

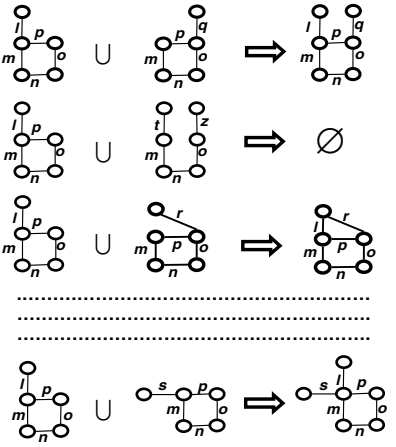


Fig. 2. Attempts to combine *lmnop* with other *5*-edge subgraphs of (L_5)

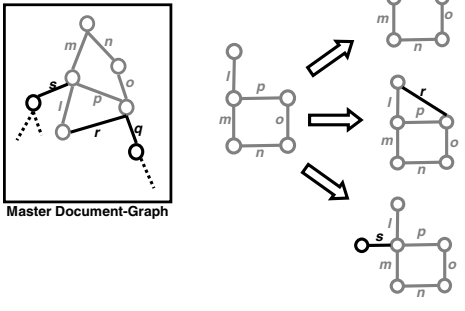


Fig. 3. *6*-edge Subgraph-Extension Generation for the *5*-edge subgraph *lmnop*

avoid comparing a large number of hash-codes for all candidates using the FSG approach. We have reduced this number of comparisons by a significant degree by implementing our own and new Subgraph-Extension Generation (SEG) method.

2) SEG: Rather than trying a brute-force strategy of comparing all possible combinations (e.g., FSG [2]), we use the MDG as the source of background knowledge to entirely eliminate the unsuccessful attempts while generating (k+1)-edge candidate subgraphs from k-edge subgraphs. We maintain a neighboring-edges' list for each k-edge subgraph and generate candidates for frequent higher order subgraphs by taking edges only from this list.

Fig. 3 shows the SEG mechanism for subgraph lmnop, which can be compared with the FSG approach of Fig. 2. The gray edges of Fig. 3 are the edges of the 5-edge subgraph which we want to extend to generate 6-edge candidates. The black lines indicate the neighboring edges which extend the 5-edge gray subgraph maintained in our MDG. The same instance is used for both Fig. 2 and Fig. 3 for an easy comparison. The neighboring-edges' list of lmnop contains edges {q, r, s}. Unlike in Fig. 2, in the example presented in Fig. 3 the SEG technique does not try to blindly generate higher order subgraphs 20 times. Rather, it proceeds only three times, using the constraints coming from knowledge about the neighboring edges of lmnop in the MDG. It results in only three attempts to generate higher-order candidate subgraphs. None of these attempts fails to generate a candidate subgraph because the mechanism depends on the physical evidence of possible extension. Therefore, the SEG offers a novel knowledge-based mechanism that eliminates unnecessary attempts to combine subgraphs. All the generated candidate subgraphs that pass the minimum support threshold are entered into a subgraph-document matrix (analogous to term-document matrix of the vector-space model of document clustering). The subgraph-document matrix is used in the document clustering process later.

3) MaxSEG: In this approach, we keep only the largest frequent subgraphs and remove all smaller subgraphs if they are contained in the higher order subgraphs. Any k-edge subgraph with support s is removed from the subgraph-document matrix if every (k+1)-edge frequent subgraph generated from it has the same support, which we will denote with s. If the k-edge subgraph generates at least one (k+1)-edge frequent subgraph that has frequency within [Sup_{MIN}, s) range, then we keep both the k-edge subgraph and the generated (k+1)-edge subgraphs.

In our implementation the SEG and the MaxSEG both require the same number of attempts to generate (k+1)-edge subgraphs from a k-edge subgraphs, but they result in different numbers of subgraphs in the subgraph-document matrix. Consider that the 5-edge subgraph lmnop of the example given in Fig. 3 appears in 20 document-graphs (support=20) and the Sup_{MIN} threshold is 10. If every generated 6-edge subgraph of the example has support=20, then we remove lmnop from the subgraph-document matrix and keep only the newly generated 6-edge subgraphs. Now consider another situation where one of the generated 6-edge subgraphs lmnopq has support=15 and the other 6-edge subgraphs have support=20. In this case, all of the 5-edge and 6-edge subgraphs lmnop, lmnopq, lmnopr and lmnops will remain in the subgraph-document matrix according to our MaxSEG approach. Therefore, the decision whether a k-edge subgraph will remain in the subgraph-document matrix or not is taken after generating all the (k+1)-edge subgraphs from this particular k-edge subgraph. This is why both

SEG and MaxSEG approaches require the same number of attempts to generate higher order subgraphs, but the numbers of their resultant subgraphs are different.

3.3 Clustering and Evaluation

GDClust [1] uses Hierarchical Agglomerative Clustering (HAC) [21] to gradually group the documents. We have chosen HAC because it facilitates the evaluation of our results from a broad range of generated clusters. The clustering unit constructs a dissimilarity matrix that contains dissimilarity values between every pair of document-graphs derived based on the number of common frequent subgraphs occurring in their respective document-graphs. We have used the cosine similarity measure [22] in all experiments because of its popularity in text clustering.

To evaluate the clustering of GDClust, we used both unsupervised and supervised evaluations of cluster validity. As an unsupervised evaluation we used the Average Silhouette Coefficient (ASC) [23]. We used entropy, purity and F-measure as supervised measures of cluster evaluation [24]. We compared our sense-based system with the *traditional* (i.e., bag-of-nouns) vector space model (VSM) that applies logarithmic normalization of keywords' frequency (i.e., TF-IDF). Additionally, we compared our sense-based system with a VSM utilizing the background knowledge of WordNet (i.e., bag-of-concepts). With this mechanism, which we called *concept* VSM, we expand each term into its corresponding synsets (i.e., concepts) from WordNet, and then we split the term's frequency between these synsets to obtain synset frequency values. Using these frequencies we compute values which are analogous to TF-IDFs in the *traditional* VSM. The same hierarchical agglomerative clustering algorithm as GDClust is used in both *traditional* and *concept* VSM-based systems to keep our experiments comparable.

4 Experimental Results and Conclusions

In our experiments, we used all 19997 documents of the 20 Newsgroups (20NG) [25] dataset. Some of the 20 class labels of the dataset can be combined together to form a higher level group. As a result, the class labels provided with the dataset may not match the clusters well and therefore our supervised evaluation must carry some imperfection. Table 1 shows the list of the 20NG, partitioned to 6 classes more or less according to subject matter as recommended in [25]. We used these 6 classes for our supervised evaluation of clustering.

Table 1. 20 Newsgroups (20NG) dataset [25]

Class	# of Docs	Original Newsgroups' Labels
1	5000	comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x
2	4000	rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey
3	4000	sci.crypt, sci.electronics, sci.med, sci.space
4	3000	talk.politics.misc, talk.politics.guns, talk.politics.mideast
5	2997	talk.religion.misc, alt.atheism, soc.religion.christian
6	1000	misc.forsale

4.1 Performance and Accuracy Analysis of the Subgraph Discovery Process

This section provides the experimental results of GDClust using the original FSG, SEG, and MaxSEG approaches. Since the FSG approach is very slow compared to our SEG and MaxSEG mechanisms, we used only a subset of 5000 documents to show the comparison between approaches (Fig. 4). We constructed the stratified subset of 5000 documents by randomly selecting 25% of the documents from each of the 6 groups of Table 1. We show results with the complete 20NG dataset in all remaining experiments.

The new SEG and MaxSEG approaches for the frequent subgraph discovery process outperform our original FSG-based strategy. Due to the speed of SEG and MaxSEG, the lines drawn for them appear linear and flat in comparison to the light-gray line of the FSG approach (Fig. 4.a), although the actual behaviors of SEG and MaxSEG are not linear (Fig.4.b). All curves maintain their hat-like shape, typical of the Apriori approaches, but due to the scale necessary to show the FSG results it is not clearly visible in Fig. 4.a.

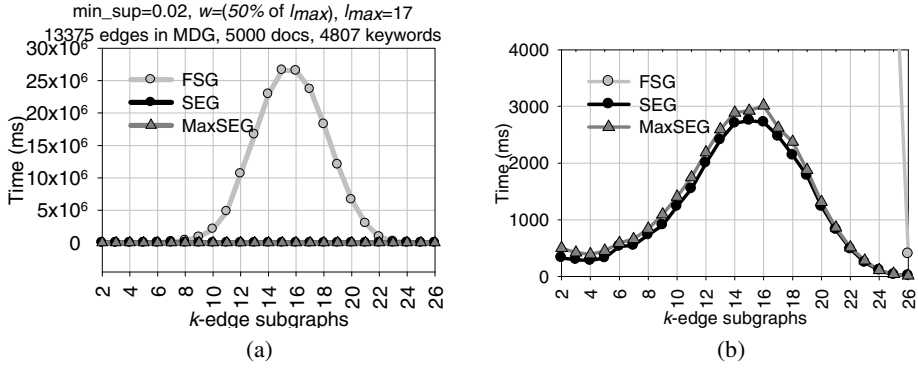


Fig. 4. Comparison between FSG, SEG, and MaxSEG via the time spent on generating k-edge subgraphs. (b) is a close-up of (a) to better show the behaviors of SEG and MaxSEG.

The SEG and MaxSEG approaches of GDClust outperform the FSG approach by a high magnitude due to the lower number of attempts used to generate higher order subgraphs by avoiding blind attempts. Table 2 shows that in every case SEG or MaxSEG saved a huge percentage of blind attempts generated by the FSG approach. The SEG and MaxSEG approaches saved 98.8% of the attempts while generating *15*-edge subgraphs from frequent *14*-edge subgraphs. Since *14*-edge subgraphs are the most frequent ones (Table 3), obviously the number of attempts to construct 15-edge subgraphs from *14*-edge subgraphs reaches the maximum for the FSG approach.

The numbers of the detected frequent *(k+1)*-edge subgraphs are the same for FSG and SEG because both methods generate complete sets of frequent subgraphs. However, they use different mechanisms to construct higher order subgraphs with different numbers of attempts. MaxSEG also generates same number of *(k+1)*-edge candidate subgraphs from the list of *k*-edge subgraphs during its execution (Table 2). But after extending a *k*-edge subgraph, MaxSEG removes it from the subgraph-document

Table 2. Number of *attempts* to generate *k+1*-size candidates from the *k*-size freq. subgraphs, where *k* is the number of iteration in Apriori.

k	FSG	SEG & MaxSEG	Savings [%]
1	---	---	---
2	11990	184	98.47%
3	7656	289	96.23%
4	6320	406	93.58%
5	8742	654	92.52%
6	18360	1150	93.74%
7	41412	2006	95.16%
8	92112	3386	96.32%
9	195806	5532	97.17%
10	416670	8954	97.85%
11	827190	13850	98.33%
12	1441200	19855	98.62%
13	2124306	25973	98.78%
14	2661792	31117	98.83%
15	*2861172*	*34365*	*98.80%*
16	2697806	35341	98.69%
17	2263520	34106	98.49%
18	1649940	30565	98.15%
19	1031240	25268	97.55%
20	539490	19035	96.47%
21	231842	12952	94.41%
22	78680	7810	90.07%
23	20306	4098	79.82%
24	3782	1827	51.69%
25	462	665	34.63%
26	30	185	23.33%

Table 3. Numbers of Frequent Subgraphs, actually discovered in each iteration (*k*) of all 3 implementations: FSG, SEG, and MaxSEG

k	FSG & SEG	Max SEG	Fr. Subgr. Reduced by MaxSEG [%]
1	110	86	21.82%
2	88	72	18.18%
3	80	68	15.00%
4	94	81	13.83%
5	136	112	17.65%
6	204	164	19.61%
7	304	236	22.37%
8	443	338	23.70%
9	646	469	27.40%
10	910	605	33.52%
11	1201	733	38.97%
12	1458	832	42.94%
13	1632	877	46.26%
14	*1692*	*846*	*50.00%*
15	1643	782	52.40%
16	1505	696	53.75%
17	1285	590	54.09%
18	1016	468	53.94%
19	735	350	52.38%
20	482	241	50.00%
21	281	150	46.62%
22	143	83	41.96%
23	62	40	35.48%
24	22	16	27.27%
25	6	5	16.67%
26	1	1	0.00%

matrix if this particular *k*-edge subgraph has generated all *(k+1)*-edge subgraphs having the same support as this *k*-edge subgraph. This is the reason why numbers of generated left frequent subgraphs are different in MaxSEG approach than the FSG or SEG (Table 3). Fig. 4.b shows that the SEG approach is slightly faster than the MaxSEG approach. This is due to the fact that MaxSEG requires additional checks to verify the supports of the newly generated *(k+1)*-edge subgraphs in order to remove their immediate *k*-edge parent subgraph. Table 3 shows the exact number of the detected frequent *k*-edge subgraphs by the SEG and the number of maximum subgraphs detected by the MaxSEG. It also shows the percentage of the subgraphs removed by the MaxSEG approach. It shows that the MaxSEG approach removes 50% of the 14-edge frequent subgraphs during the generation of *15*-edge subgraphs.

Although the MaxSEG approach removes a lot of smaller subgraphs, our results show that it does not cause the same decrease of clustering quality. Table 4 shows that FSG, SEG and MaxSEG result in much better clustering than the VSMs. Note that

Table 4. Dissimilarity Matrix construction times and the results of hierarchical agglomerative clustering of the 20NG dataset, presented via: Entropy, Purity, F-measure and Average Silhouette Coefficient

	Time (sec)	Supervised Evaluations			Unsupervised Eval. via ASC
		Entropy	Purity	F-meas.	
FSG	768	0.84	0.81	0.77	0.76
SEG	768	0.84	0.81	0.77	0.76
MaxSEG	462	0.87	0.81	0.76	0.63
Traditional VSM	160	2.46	0.26	0.28	0.04
Concept VSM	95	2.46	0.25	0.26	-0.07

while SEG offers faster execution than FSG, the clustering results are the same since they both generate the same set of subgraphs. Additionally, although the MaxSEG approach removes approximately 45% of all discovered frequent subgraphs, it is not penalized during the clustering process in this ratio. Table 4 shows that the F-measure for MaxSEG is just slightly lower than the F-measure of the SEG and FSG approaches. Since MaxSEG reduces the number of subgraphs, it offers faster construction of the dissimilarity matrix during the clustering phase compared to the FSG and SEG driven approaches (see column *Time (sec)* in Table 4).

Our GDClust-based approach to document clustering shows more accurate results than the vector space models of document clustering. Table 4 shows that the best clustering is found using the SEG approach. The traditional VSM based approach does not show good groupings of data. ASC=0.04 for 6 clusters indicates that the VSM fails to provide clear separation between clusters. Furthermore, the concept-based VSM provides even worse separation between clusters, which indicates that the inclusion of background knowledge alone is not enough to provide good results.

4.2 Conclusions

GDClust presents a valuable technique for clustering text documents based on the co-occurrence of frequent senses in documents. The approach offers an interesting, sense-based alternative to the commonly used VSM for clustering text documents. Unlike traditional systems, GDClust harnesses its clustering capability from the frequent senses discovered in the documents. It uses graph-based mining technology to discover frequent senses. Unlike chemical compounds, our document-graphs may contain thousands of edges which results in a slow generation of frequent subgraphs during the discovery process using pre-existing graph mining techniques. We have introduced the SEG and the MaxSEG techniques of frequent subgraph generation, which outperform the previous used FSG strategy by a high magnitude by taking advantage of the constraints coming from our knowledge about natural-language. We have shown that our proposed approaches perform more accurately than VSMs.

References

1. Hossain, M.S., Angryk, R.: GDClust: A Graph-Based Clustering Technique for Text Documents. In: IEEE ICDM 2007, Workshop on Mining Graphs and Complex Structures, pp. 417–422 (2007)

2. Kuramochi, M., Karypis, G.: An efficient algorithm for discovering frequent subgraphs. IEEE Trans. on KDE 16(9), 1038–1051 (2004)
3. Sebastiani, F.: Machine learning in automated text categorization. ACM Comp. Surveys 34(1), 1–47 (2002)
4. Manning, C.D., Schutze, H.: Foundations of NLP. MIT Press, Cambridge (1999)
5. Cleverdon, C.: Optimizing convenient online access to bibliographic databases. Information Survey and Use 4(1), 37–47 (1984)
6. Cognitive Science Laboratory Princeton University, WordNet: A Lexical Database for the English Language, http://wordnet.princeton.edu/
7. Miller, G., Beckwith, R., FellBaum, C., Gross, D., Miller, K., Tengi, R.: Five papers on WordNet. Princeton University, Princeton (1993)
8. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proc. of the 20th Intl. Conf. on Very Large Data Bases, pp. 487–499 (1994)
9. Chittimoori, R.N., Holder, L.B., Cook, D.J.: Applying the SUBDUE substructure discovery system to the chemical toxicity domain. In: 12th FLAIRS Conf. 1999, pp. 90–94 (1999)
10. Piriyakumar, D.A.L., Levi, P.: An efficient A* based algorithm for optimal graph matching applied to computer vision. In: GRWSIA 1998 (1998)
11. Dupplaw, D., Lewis, P.H.: Content-based image retrieval with scale-spaced object trees. In: SPIE: Storage and Retrieval for Media Databases, vol. 3972, pp. 253–261 (2000)
12. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45(2), 167–256 (2003)
13. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: ICDM 2002, pp. 721–724 (2002)
14. Moti, C., Ehud, G.: Diagonally Subgraphs Pattern Mining. In: 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 51–58 (2004)
15. Ketkar, N., Holder, L., Cook, D., Shah, R.: Subdue: Compression-based Frequent Pattern Discovery in Graph Data. In: KDD Workshop on Open-Source Data Mining, pp. 71–76 (2005)
16. Agrawal, R., Mehta, M., Shafer, J., Srikant, R., Arning, A., Bollinger, T.: The Quest Data Mining System. In: KDD 1996, pp. 244–249 (1996)
17. Mannila, H., Toivonen, H., Verkamo, I.: Efficient Algorithms for Discovering Association Rules. In: AAAI Workshop on KDD, pp. 181–192 (1994)
18. Han, J., Fu, Y.: Discovery of multiple-level association rules from large databases. In: 21st Intl. Conf. on VLDB, pp. 420–431 (1995)
19. Tomita, J., Kikui, G.: Interactive Web search by graphical query refinement. In: 10th Intl. WWW Conf., pp. 190–191 (2001)
20. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid Prototyping for Complex Data Mining Tasks. In: Intl. Conference on KDD, pp. 935–940 (2006)
21. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: SIGMOD Conf. on Management of Data, pp. 103–114 (1996)
22. White, R., Jose, J.: A study of topic similarity measures. In: 27th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 520–521 (2004)
23. Lin, F., Hsueh, C.M.: Knowledge map creation and maintenance for virtual communities of practice. Information Processing and Management: an International Journal 42(2), 551–568 (2006)
24. Tan, P.N., Steinbachm, M., Kumar, V.: Introduction to Data Mining, pp. 539–547. Addison-Wesley, Boston (2005)
25. Rennie, J.: Homepage for 20 Newsgroups Dataset, http://people.csail.mit.edu/jrennie/20Newsgroups/

A Taxonomic Generalization Technique for Natural Language Processing

Stefano Ferilli[1,2], Nicola Di Mauro[1,2], Teresa M.A. Basile[1], and Floriana Esposito[1,2]

[1] Dipartimento di Informatica – Università di Bari
{ferilli,ndm,basile,esposito}@di.uniba.it
[2] Centro Interdipartimentale per la Logica e sue Applicazioni – Università di Bari

Abstract. Automatic processing of text documents requires techniques that can go beyond the lexical level, and are able to handle the semantics underlying natural language sentences. A support for such techniques can be provided by taxonomies that connect terms to the underlying concepts, and concepts to each other according to different kinds of relationships. An outstanding example of such a kind of resources is WordNet. On the other hand, whenever automatic inferences are to be made on a given domain, a generalization technique, and corresponding operational procedures, are needed. This paper proposes a generalization technique for taxonomic information and applies it to WordNet, providing examples that prove its behavior to be sensible and effective.

1 Introduction

Due to the complexity of natural language, most NLP techniques in the literature have so far focused on lexical-level representations such as bags of words. Unfortunately, using a strictly syntactical approach to text processing is often insufficient, because the words make direct and explicit reference to underlying concepts that have complex interactions and relationships to each other, and that are fundamental to understand the text and to relate texts to each other. For instance, two sentences such as "the man bought a car" and "the woman won a bicycle" would be considered as having nothing in common, while any human would immediately grasp their generalization of "a person acquiring a means of transportation". Hence, the need to introduce and exploit taxonomic knowledge, as provided by existing lexical resources and ontologies. In this work, we will focus on the exploitation of taxonomic resources to set up a generalization framework. This involves two components: a procedure that, given two (sets of) words or concepts returns their generalization, and a procedure that, given a taxonomic model (possibly coming from previous generalizations of words/concepts) and an observed word/concept, checks whether the former covers the latter. In this paper, we will use a widely-known general-purpose lexical taxonomy, WordNet, as a sample resource on which demonstrating the proposed technique. However, it should be noted that this decision is just driven by the opportunity of having a wide-scope, readily-available taxonomic resource for testing the technique. Indeed, the technique itself is completely general and can be applied to

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 418–427, 2011.

any other (possibly domain-specific) resource having very general features provided by WordNet: the generalization-specialization relationship (for nouns and verbs) and some kind of relationship expressing closeness or similarity among taxonomic items (such as synonymy).

After providing some background and fundamental notions in the next section, Section 3 describes the approach to generalization and coverage, while Section 4 provides a qualitative validation of the approach. Finally, Section 5 concludes the paper and proposes future work issues.

2 Background

This section discusses the basic features of an ontology that are needed to apply the generalization technique proposed in the following sections. Although these features are general, they will be framed in the WordNet environment to have a more practical example. WordNet [1, 2] is a famous lexical taxonomy/ontology[1] inspired by psycho-linguistic theories on human memory. It takes into account two main concepts: *word forms*, i.e. their written aspect, and *word meanings*, i.e. their underlying concept. Differently from classical dictionaries, terms in WordNet are not organized as an alphabetically ordered list, but arranged in a graph determined by various kinds of relationships. Nodes representing terms are linked to the nodes representing the corresponding meanings. From the opposite perspective, each node representing a meaning is connected to all synonymous words expressing that meaning, this way defining the fundamental concept of *synset* ('synonymous set'). Synsets can be considered as unique identifiers for meanings (in the rest of this paper, the terms 'concept', 'meaning', 'sense' and 'synset' will be used interchangeably), and a textual definition, called a *gloss*, is also provided for each of them. Clearly, due to polysemy one term might have different meanings (i.e., be associated to many synsets), and might even belong to different syntactic categories. The current version of WordNet (3.0) includes more than 150.000 lexical forms and approximately 120.000 synsets.

Two kinds of relationships are defined in WordNet. Semantic ones always relate two synsets/meanings, while lexical relationships, on the other hand, involve both terms and synsets [3]. For the purposes of our technique, we need to focus just on the following semantic relationships:

Hyperonymy determines a generalization hierarchy on synsets, and is defined on nouns and verbs. It links a synset X to a more general one Y such that "X is a kind of Y". Interpreting it the other way round, one obtains its opposite relationship, *hyponymy*, that links a concept Y to a more specialized one X and hence determines specialization hierarchies. They are the largest relationships in WordNet.

Similarity used among adjectives only, according to the relationship of antinomy. The main adjectives on which such a relationship is set are called *head*

[1] There is a debate about the latter definition, since some require an ontology to contain formal definitions of the concepts.

synsets, and in turn are connected to similar *satellite synsets* that somehow inherit the antinomy relationships from the main meaning to which they are connected.

and on the following lexical ones:

Sinonymy is, as already pointed out, the main relationship in WordNet. Interestingly, synonymous terms are not directly connected to each other, but they are connected to the synset representing the underlying meaning. Thus, two terms are implicitly synonyms if both are linked to the same synset. Among several possible definitions of synonymy, WordNet adopts a perspective according to which two terms are synonyms if they can be safely replaced to each other in a given linguistic context without changing the sentence meaning. This clearly avoids the possibility of two terms in different syntactic categories being synonymous, and in practice neatly divides the whole taxonomy into four separate parts, corresponding to the syntactic categories of nouns, verbs, adjectives and adverbs.

See also a lexical relationship connecting related terms such that the latter helps in defining or understanding the former.

Of course, availability of additional relationships in the taxonomy, although not required, can allow to extend and refine the proposed technique if suitably exploited.

WordNet has gained a lot of attention in the literature, as a wide-coverage, general-purpose linguistic resource that tries to bridge the gap between the lexical level and the underlying semantics. Several translations (both in different languages and cross-language), tailorings to specific application domains, and extensions with additional information (e.g., WordNet Domains [4]) have been carried out. It has been thoroughly exploited for many tasks, among which Word Sense Disambiguation [5] and similarity assessment among concepts [6, 7, 8]. However, not much work seems to be available concerning inference strategies defined on the WordNet taxonomy to exploit it as a support for reasoning about (terms,) concepts and their relationships.

3 Taxonomic Generalization and Coverage

The problem we will face in this work can be defined as follows: given two concepts in a WordNet-like taxonomy, how to define their generalization. For instance, such a generalization may act as a model to be checked against further observed concepts. Hence, the problem of how to assess whether a model accounts for an observation. This setting can be extended taking into account sets of concepts instead of single concepts. This allows to handle words, even without any hint about their grammatical role and exact meaning, by replacing them with the set of their possible associated concepts in any grammatical category. In WordNet, nouns/verbs on one hand, and adjectives/adverbs on the other, have a very similar organization and relationships involving them. Accordingly, we will devise two generalization/coverage strategies.

For nouns and verbs a hierarchical generalization based on the Hyponymy relationship is implicitly available. The ancestors of a concept according to such a relationship can be interpreted as all its possible generalizations. A classical approach has been to take as a generalization of given concepts their Least Common Subsumer (i.e., the closest common ancestor). This might be misleading when the taxonomy is actually a heterarchy, where there might be several incompatible Least Common Subsumers. An alternative naive approach might consist in taking as a generalization of two synsets the set of all their common ancestors, and in saying that a model of this kind covers an observation term/synset if and only if there is a non-empty intersection between the model and the set of ancestors of the observation. However, this would be very loose, because there is a highest chance that the top (i.e., the most general and abstract) concepts in the hierarchy appear in both, resulting in the coverage of almost everything. Indeed, in such a hierarchy the closest concept that generalizes two given concepts is almost always too general to be of use (often just the root of the hierarchy, 'Entity') and might not be unique (due to polysemy). The problem is that the top-level ancestors are very general, and hence useless in practice because they would be over-generalizations. A solution might be ignoring (i.e., removing from the hierarchy) the very top concept 'Entity', or even the highest levels in the hierarchy, but this would introduce the problem of how to determine up to which level to ignore, and how being sure that the removed levels are in fact irrelevant to the task correctness. To avoid this problem, a baseline approach that we will adopt consists in assuming that a model covers an observation if and only if the ancestors of the observation include all of the model synsets. It is clearly a very cautious/pessimistic approach (requiring that the set of synsets in the model is a subset of the set of ancestors of the observation is a very strict bias), but can serve as a reference for comparing the performance of other solutions.

We propose a generalization technique that selects, among all common ancestors of the two elements to be generalized, the 'border' set of all 'minimal' ancestors (in the sense that they have no descendant in the set of common ancestors, a kind of leaves of such a hierarchy). Given a set of concepts X, let us denote by B_X the border of X, and by A_X the remaining ancestors of X. In some sense, the border represents the set of all minimally general generalizations, resembling for this the version space approach [9]. Considering only the 'border' subset, the model is not more general than needed and hence does not include concepts more general than those it should account for. In this way, the initial option of checking for a non-empty intersection between the ancestors of the observation (including the synsets in the observation itself) and the border synsets in the model becomes much more sensible. The underlying rationale is that the model specificity is expressed by its border synsets, and not by all of its ancestors, and hence only the former should be exploited for checking observation coverage. More formally: given a model M and an observation O, M *covers* O if $B_M \cap (B_O \cup A_O) \neq \emptyset$, i.e. at least one of the meanings in the model covers (i.e., is in the generalization hierarchy of) one of the synsets in the observation. Using B_M (instead of $B_M \cup A_M$, as in the naive version) avoids that

Algorithm 1. Generalization technique for nouns and verbs in a taxonomy

Require: X: set of concepts
Require: T: taxonomy
  $A \leftarrow$ concepts(T) /* set of common ancestors */
  for all $x \in X$ **do**
    $A \leftarrow A \cap \{a \in \text{concepts}(T) \mid \text{ancestor}_T(a, x)\}$
  end for
  return $\{c \in A \mid \nexists c' \in A \text{ s.t. } \text{ancestor}_T(c', c)\}$

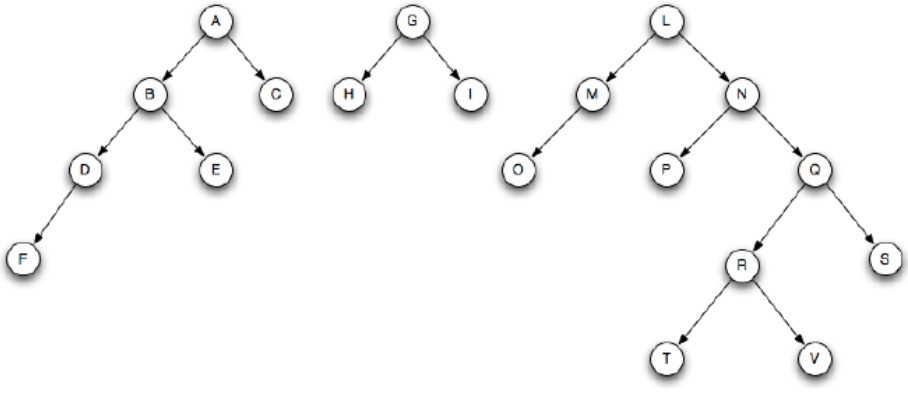


Fig. 1. A hypothetical fragment of taxonomy

the observation is covered by an ancestor in the model and hence, actually, by an over-generalization, and with respect to the baseline model is not so strict as to require that all of the model synsets are included in the ancestors of the observation. Algorithm 1 sketches the pseudo-code of the procedure, assuming a taxonomy on whose elements an 'ancestor' relationship is defined (corresponding to the transitive closure of the generalization relationship).

As an example, consider the tree in Figure 1, and two words P_1 and P_2, corresponding respectively to synsets (nodes) $\{F, H, T\}$ and $\{E, I, S\}$. Their generalization M is equal to $\{B, A, G, Q, N, L\}$, where the border of M is $B_M = \{B, G, Q\}$ and the set of ancestors of M is $A_M = \{A, N, L\}$. This is the chosen model. Now, let us consider word P_3, corresponding to the set of synsets $\{D, R\}$: the set $(B_{P_3} \cup A_{P_3})$ including both its nodes and the ancestors of its nodes is $\{D, B, A, R, Q, N, L\}$. Thus, since $(B_{P_3} \cup A_{P_3}) \cap B_M \neq \emptyset$, P_3 is covered by M. Finally, consider word P_4, corresponding to the set of nodes $\{C, P\}$. The set $(B_{P_4} \cup A_{P_4})$ comprising its nodes and their ancestors is $\{C, A, P, N, L\}$. Since $(B_{P_4} \cup A_{P_4}) \cap B_M = \emptyset$, P_4 is not covered by M.

As another example actually taken from WordNet, generalizing 'pencil' and 'rubber' (both can be interpreted both as tools and as the underlying matter), the baseline generalization would be:

[entity], [whole,whole_thing,unit], [object,physical_object], [substance,matter], [artifact,artefact], [implement], [instrumentality,instrumentation]

whose border is:

[substance,matter], [implement]

Adjectives are not hierarchically organized as nouns and verbs. E.g., although 'colored' can be considered as a generalization of 'red', such a relationship is not specified in WordNet. The following relationships are available in WordNet for adjectives: Similarity, Attribute, See also, Participial, Pertinence, Derivation. We propose to select the set of items connected by the 'Similarity' and 'See also' relationships to the two adjectives to be generalized, because in a sense they express all possible variations thereof, and then taking their intersection. Since in this case the generalization does not consist of more abstract concepts, but of closely related ones, the coverage algorithm for adjectives consists in checking that there is a non-empty intersection between the adjectives in the model and the set of 'ways for defining' the corresponding synset in the observation. I.e., at least an element in the model must be related by similarity (relationship 'Similarity') or must provide further information (through relationship 'See also') to the adjective synset in the observation. This strategy can be applied also to adverbs derived from adjectives, by switching to the corresponding adjectives, while no hint is available for the others.

4 Evaluation

To evaluate the viability of the proposed techniques, several groups of 6 words for each word category were chosen from WordNet3.0 by linguistic experts, to which a concept of term generalization was provided, but who were not aware of the specific algorithm discussed above. Specifically, for each group they selected two reference words to be generalized, and four more test words: two somehow related to the reference words (that, in principle, had to be covered by the generalization), and two conceptually unrelated (that should not be covered by the generalization). This setting was devised to provide indications of the generalization and coverage behavior on both false positives and false negatives, for evaluating both completeness and consistency. The performance of the generalization was then compared to the naive baseline. In the following, for each group of test words (along with an explanation for the choice of such words, when useful), a table will show the behavior of the proposed technique against the baseline (where Y means that the generalization covers the observation, while N means that it does not).

First, a few cases taken from the category of nouns were considered. In Case 1, there are two tricky tests: 'boy', that is to be covered with reference to an application of the generalized terms to persons, and 'antilope', that might be misleading being an animal, but must not be covered being a herbivore.

Case 1 : dog-cat

Term	theoretical	motivation	proposal	baseline
leopard	Y	a carnivore	Y	N
boy	Y	cat and dog are used also referred to persons	Y	N
antilope	N	a herbivore	N	N
book	N		N	N

Case 2 : nail-hammer

Term	theoretical	motivation	proposal	baseline
hand	Y	nail and hammer (a bone in the ear) are parts of the body	Y	N
saw	Y	nail and hammer are carpentry tools	Y	N
girl	N		N	N
river	N		N	N

Some examples from the category of verbs are reported below. Specifically, Case 2 is particularly interesting due to the highly polysemic behavior of 'play'.

Case 1 : introduce-repose

Term	theoretical	motivation	proposal	baseline
insert	Y	putting something in	Y	N
enclose	Y	putting something in a container	Y	N
change	N		N	N
increase	N		N	N

Case 2 : play-represent

Term	theoretical	motivation	proposal	baseline
pretend	Y	indicates an artificial behavior, like in drama acting	Y	N
perform	Y	to put on a show or performance	Y	N
swim	N		N	N
think	N		N	N

Then, for adjectives, words that in at least one of their meanings can be considered similar to the reference pair were taken into account as positive test cases. Interestingly, in Case 1 coverage is correctly recognized also for test cases that refer to different perspectives on the reference terms ('incisive' is an abstract interpretation, while 'needlelike' is more concrete).

Case 1 : sharp-acute

Term	theoretical	motivation	proposal	baseline
incisive	Y	referred to the effectiveness of a reasoning, way of thinking or of speaking	Y	Y
needlelike	Y	somehow in-between the two reference terms	Y	Y
happy	N	a state-of-mind	N	N
hungry	N	a psycho-physical state	N	N

Case 2 : tiny-little

Term	theoretical	motivation	proposal	baseline
young	Y	has a similar meaning to the reference terms when referred to persons	N	N
small	Y	has a similar meaning to the reference terms when referred to persons or things	Y	Y
depressed	N	a state-of-mind	N	N
long	N	referred to distance rather than size	N	N

Finally, some adverbs derived from adjectives were considered, using the same rationale as reported above for adjectives.

Case 1 : quickly-rapidly

Term	theoretical	motivation	proposal	baseline
speedily	Y	a synonym	Y	N
apace	Y	a synonym	Y	N
near	N		N	N
easily	N		N	N

Case 2 : appropriately-fittingly

Term	theoretical	motivation	proposal	baseline
suitably	Y	a synonym	Y	N
fitly	Y	a synonym	Y	N
jointly	N		N	N
playfully	N		N	N

Table 1 summarizes the results for the complete set of word groups, and for each word category, both overall and on positive/negative cases only. It clearly emerges that the proposed solution not only represents an outstanding improvement on the baseline, but produces very high-quality results in itself. More specifically, although the generalizations are more compact in the proposed procedure due to the focus on the border only, the key for the improvement is represented by the coverage procedure. Indeed, as to nouns and verbs, the baseline coverage strategy is very pessimistic, and rejects almost all test words: all negative cases are correctly rejected, but no positive cases are covered for nouns, nor for verbs. Conversely, the proposed technique provides a perfect behavior on both positive and negative cases for these categories. On adjectives and adverbs the generalization and coverage strategies are the same for both the 'baseline' and the 'proposal', but two different evaluation strategies were compared: in the former, pairwise comparisons are carried out among single meanings in the two references, while the latter adopts a global approach that first expands all meanings, and only subsequently intersects the resulting sets as a whole. Actually, on adverbs the proposed approach turns out to be (significantly) better. Conversely,

Table 1. Statistics on performance (accuracy) of the taxonomic generalization and coverage procedures

Coverage	Proposal		Baseline		Improvement
All	31/32	97%	19/32	59%	38%
positive only	15/16	94%	3/16	19%	75%
negative only	16/16	100%	16/16	100%	0%
Nouns	8/8	100%	4/8	50%	50%
positive only	4/4	100%	0/4	0%	100%
negative only	4/4	100%	4/4	100%	0%
Verbs	8/8	100%	4/8	50%	50%
positive only	4/4	100%	0/4	0%	100%
negative only	4/4	100%	4/4	100%	0%
Adjectives	7/8	87%	7/8	87%	0%
positive only	3/4	75%	3/4	75%	0%
negative only	4/4	100%	4/4	100%	0%
Adverbs	8/8	100%	4/8	50%	50%
positive only	4/4	100%	0/4	0%	100%
negative only	4/4	100%	4/4	100%	0%

as to adjectives, the coverage performance of the proposed approach is just the same as the baseline, although it should be said that it could be hardly improved because only one error on positive cases occurs.

5 Conclusions

Several techniques for processing texts are based on the lexical level in order to make the problem computationally tractable. However, the tricks of natural language, and the relationships among the concepts underlying the words that are used in text, call for some kind of taxonomic background knowledge to help handling the semantics underlying the sentences. An outstanding example of such a kind of resources is WordNet. This paper proposed a generalization procedure, and a coverage procedure, to be exploited for automatic inferences on a given domain for which basic taxonomic information is provided. Although the problem is not very suitable to statistic evaluation, selected test cases were devised, and the outcome of the proposed technique on WordNet terms revealed that its behavior is sensible and effective.

Future work, part of which is already ongoing, includes running wider and more varied experiments. Moreover, further relationships expressed in WordNet might be exploited for improving the generalization. The proposed technique might be exploited to extend a purely syntactical approach to inference in First-Order Logic, where some predicates are not just uninterpreted syntactic entities, but are associated to nodes in a taxonomy, which would allow to exploit the relationships in the taxonomy as a background knowledge in order to tackle more complex problems. In particular, we intend to embed it in an existing

framework for symbolic Machine Learning presented in [10, 11]. A possible example of application might be learning from structural representations of natural language sentences, as proposed in [12].

References

[1] Miller, G.A., Beckwith, R., Fellbaum, C., Miller, K., Gross, D.: Introduction to wordnet: An on-line lexical database. International Journal of Lexicography 3(4), 235–244 (1990)

[2] Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)

[3] Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM 38, 39–41 (1995)

[4] Bentivogli, L., Forner, P., Magnini, B., Pianta, E.: Revising the wordnet domains hierarchy: semantics, coverage and balancing. In: Proceedings of the Workshop on Multilingual Linguistic Ressources, MLR 2004, pp. 101–108. Association for Computational Linguistics, Stroudsburg (2004)

[5] Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)

[6] Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research 11, 93–130 (1999)

[7] Budanitsky, A., Hirst, G.: Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In: Proc. Workshop on WordNet and Other Lexical Resources, 2nd Meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh (2001)

[8] Ferilli, S., Biba, M., Di Mauro, N., Basile, T.M., Esposito, F.: Plugging taxonomic similarity in first-order logic horn clauses comparison. In: Serra, R., Cucchiara, R. (eds.) AI*IA 2009. LNCS, vol. 5883, pp. 131–140. Springer, Heidelberg (2009)

[9] Mitchell, T.M.: Version spaces: a candidate elimination approach to rule learning. In: Proceedings of the 5th International Joint Conference on Artificial intelligence, vol. 1, pp. 305–310. Morgan Kaufmann Publishers Inc., San Francisco (1977)

[10] Semeraro, G., Esposito, F., Malerba, D., Fanizzi, N., Ferilli, S.: A logic framework for the incremental inductive synthesis of datalog theories. In: Fuchs, N. (ed.) LOPSTR 1997. LNCS, vol. 1463, pp. 300–321. Springer, Heidelberg (1998)

[11] Ferilli, S., Basile, T.M.A., Biba, M., Di Mauro, N., Esposito, F.: A general similarity framework for horn clause logic. Fundamenta Informaticae 90, 43–66 (2009)

[12] Ferilli, S., Fanizzi, N., Semeraro, G.: Learning logic models for automated text categorization. In: Proceedings of the 7th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence, UK, pp. 81–86. Springer, Heidelberg (2001)

Fr-ONT: An Algorithm for Frequent Concept Mining with Formal Ontologies

Agnieszka Ławrynowicz[1] and Jedrzej Potoniec[1]

Institute of Computing Science, Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznan, Poland
{alawrynowicz,jpotoniec}@cs.put.poznan.pl

Abstract. The paper introduces a task of frequent concept mining: mining frequent patterns of the form of (complex) concepts expressed in description logic. We devise an algorithm for mining frequent patterns expressed in standard $\mathcal{EL}^{++}$ description logic language. We also report on the implementation of our method. As description logic provides the theorethical foundation for standard Web ontology language OWL, and description logic concepts correspond to OWL classes, we envisage the possible use of our proposed method on a broad range of data and knowledge intensive applications that exploit formal ontologies.

1 Introduction

One of the fundamental data mining tasks is the discovery of frequent patterns. A branch of the research in this area investigates methods for mining patterns in relational, logical representations within *Inductive Logic Programming (ILP)* [1] framework. Within the setting of ILP, frequent pattern mining has been investigated initially for Datalog language, in systems such as WARMR [2], FARMER [3] or c-armr [4]. Recently, however, with increased availability of information published using standard Semantic Web languages like OWL[1] (grounded theoretically on *description logic (DL)* [5]), new approaches are needed to mine these new relational sources of data. Therefore some recent proposals like SPADA [6] or SEMINTEC [7] have extended the scope of relational frequent pattern mining, based on ILP methodology to operate on description logic, or hybrid languages (combining Datalog with DL or DL with some form of rules). However, none of the approaches to mine frequent patterns have targeted so far peculiarities of the DL formalism, namely variable–free notation, explicit use of quantifiers (e.g. $\exists$), and DL constructors (e.g. $\sqcap$) in representing patterns.

This paper aims to fill this gap. Our main contributions are as follows: (a) a novel setting for the task of frequent pattern mining is introduced, coined *frequent concept mining*, where patterns are (complex) concepts expressed in DL (corresponding to OWL classes); (b) basic building blocks for this new setting (generality measure, refinement operator) are provided, (c) an algorithm is devised for mining frequent patterns expressed in a standard $\mathcal{EL}^{++}$ DL language. We report also on the implementation of our method.

[1] http://www.w3.org/TR/owl-features

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 428–437, 2011.

Table 1. Syntax and semantics of $\mathcal{EL}^{++}$

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$(C \sqcap D)$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$(\exists R.C)$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
concrete domain	$p(f_1,...,f_k)$ for $p \in pred(\mathbf{D})$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b_1,...,b_k \in \Delta^{\mathbf{D}}$: $f_i^{\mathcal{I}}(a) = b_i$ for $1 \leq i \leq k \wedge (b_1,...,b_k) \in p^{\mathbf{D}}\}$
GCI	$\mathcal{C} \sqsubseteq \mathcal{D}$	$\mathcal{C}^{\mathcal{I}} \subseteq \mathcal{D}^{\mathcal{I}}$
RI	$R_1 \circ \cdots \circ R_k \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \cdots \circ R_k^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
domain restriction	$\mathsf{dom}(R) \sqsubseteq \mathcal{C}$	$R^{\mathcal{I}} \subseteq C^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
range restriction	$\mathsf{ran}(R) \sqsubseteq \mathcal{C}$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C^{\mathcal{I}}$
concept assertion	$C(a)$	$a^{\mathcal{I}} \subseteq C^{\mathcal{I}}$
role assertion	$R(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \subseteq R^{\mathcal{I}}$

2 Preliminaries

2.1 Representation and Inference

Description logics [5] are a family of knowledge representation languages, equipped with a model-theoretic semantics and reasoning services. Basic elements in DLs are: *atomic concepts* (denoted by A), and *atomic roles* (denoted by R, S). *Complex descriptions* (denoted by C and D) are inductively built by using concept and role *constructors*. Furthermore, let by N_C, N_R, N_I denote the sets of *concept names*, *role names* and *individual names* respectively.

Semantics is defined by *interpretations* $\mathcal{I}$=$(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where non-empty set $\Delta^{\mathcal{I}}$ is the domain of the interpretation and $\cdot^{\mathcal{I}}$ is an interpretation function which assigns to every atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to every atomic role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to complex concept descriptions by the inductive definition as presented in Tab. 1. A DL *knowledge base*, KB, is formally defined as: $KB = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is called a TBox, and it contains axioms dealing with how concepts and roles are related to each other, and where $\mathcal{A}$ is called an ABox, and it contains assertions about individuals such as $C(a)$ (the invidual a is an instance of the concept C) and $R(a,b)$ (a is R-related to b). Moreover, DLs may also support reasoning with *concrete datatypes* such as strings or integers. A *concrete domain* $\mathbf{D}$ consists of a set $\Delta^{\mathbf{D}}$, the domain of $\mathbf{D}$, and a set $pred(\mathbf{D})$, the predicate names of $\mathbf{D}$. Each predicate name p is associated with an arity n, and an n-ary predicate $p^{\mathbf{D}} \subseteq (\Delta^{\mathbf{D}})^n$. The abstract domain $\Delta^{\mathcal{I}}$ and the concrete domain $\Delta^{\mathbf{D}}$ are disjoint. By introducing a set of *feature names* N_F, we provide a link between the DL and the concrete domain. In Table 1, by p is denoted a predicate of some concrete domain $\mathbf{D}$ and $f_1,...,f_k$ denote feature names. Within this work, we are interested in *concrete roles* $P \in N_P$ which are interpreted as binary relations $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}}$.

Example 1 provides a sample DL knowledge base, constructed based on the example of a benchmark relational financial dataset [8].

Example 1 (Description Logic KB).
$\mathcal{T}$ = { Client ≡ ∃isOwnerOf, ⊤ ⊑ ∀isOwnerOf.(Account ⊔ CreditCard), OKLoan ⊑ Loan, FinishedLoan ⊑Loan, RunningLoan ⊑Loan, FinishedLoan ≡ ¬RunningLoan }.

$\mathcal{A}$ = { isOwnerOf(Anna, a1), Account(a1), livesIn(Anna, Prague), hasAge(Anna, 35), hasLoan(a1, loan1), FinishedLoan(loan1), OKLoan(loan1), isOwnerOf(Tom, a2), Account(a2), Client(Mark), livesIn(Mark, Prague) }. □

The inference services, further referred to in the paper, are *subsumption* and *retrieval*. Given two concept descriptions C and D in a TBox $\mathcal{T}$, C subsumes D (denoted by $D \sqsubseteq C$) if and only if, for every interpretation $\mathcal{I}$ of $\mathcal{T}$ it holds that $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$. C equivalent to D (denoted by $C \equiv D$) corresponds to $C \sqsubseteq D$ and $D \sqsubseteq C$. The *retrieval* problem is, given an ABox $\mathcal{A}$ and a concept C, to find all individuals a such that $\mathcal{A} \models C(a)$.

2.2 Refinement Operators for DL

Learning in DLs can be seen as a search in the space of concepts. In ILP it is common to impose an ordering on this search space, and apply *refinement operators* to traverse it [1]. Downward refinement operators construct specialisations of hypotheses (concepts, in this context). Let $(S, \preceq)$ be a quasi ordered space. Then, a downward refinement operator ρ is a mapping from S to 2^S, such that for any $C \in S$, $C' \in \rho(C)$ implies $C' \preceq C$. C' is called a specialisation of C. For searching the space of DL concepts, a natural quasi-order is subsumption. If C subsumes D ($D \sqsubseteq C$), then C covers all instances that are covered by D. In this work, subsumption is assumed as a *generality measure* between concepts. For refinement operators proposed for DL refer further to [9,10,11,12].

3 The Task of Frequent Concept Mining

In this section, the task of frequent concept mining is formally introduced.

The definition of this task requires a specification of what is counted to calculate the pattern *support*. In the setting proposed in this paper, the support of concept C is calulated relatively to the number of instances of a user-specified concept of reference, *reference concept* $\hat{C}$, from which the search procedure starts (and which is being specialized). Importantly, for counting within the framework of DL we make the Unique Names Assumption.

Definition 1 (Support). *Let C be a concept expressed using predicates from a DL knowledge base $KB = (\mathcal{T}, \mathcal{A})$, $memberset(C, KB)$ be a function that returns the set of all individuals a such that $\mathcal{A} \models C(a)$, and let $\hat{C}$ denote a reference concept, where $\hat{C}$ is a primitive concept, and $C \sqsubseteq \hat{C}$.*

A support of pattern C with respect to the knowledge base KB is defined as the ratio between the number of instances of the concept C, and the number of instances of the reference concept $\hat{C}$ in KB: $support(C, KB) = \frac{|memberset(C,KB)|}{|memberset(\hat{C},KB)|}$. □

It is now possible to formulate a definition of *frequent concept discovery*.

Definition 2 (Frequent concept discovery). *Given i) a knowledge base KB represented in DL, ii) a set of patterns in the form of a concept C, where each C is subsumed by a reference concept $\hat{C}$ ($C \sqsubseteq \hat{C}$), iii) a minimum support threshold* minsup *specified by the user, and assuming that patterns with support s are frequent in KB if $s \geq$ minsup, the task of* frequent concept discovery *is to find the set of frequent patterns in the form of DL concepts.* □

Example 2. Let us consider the knowledge base KB from Example 1. Let us assume that $\hat{C}$=Client. There are 3 instances of $\hat{C}$ in KB. Some example patterns, refinements of Client, that could be generated are as follows:

C_1 = Client ⊓∃isOwnerOf.Account, $support(C_1, KB) = \frac{2}{3}$
C_2 =Client ⊓∃livesIn.{Prague}, $support(C_2, KB) = \frac{2}{3}$
C_3 =Client ⊓∃hasAge.$_{35}$, $support(C_3, KB) = \frac{1}{3}$
C_4 =Client ⊓∃isOwnerOf.(Account ⊓∃hasLoan.(FinishedLoan ⊓ OKLoan)) ⊓∃livesIn.{Prague} ⊓∃hasAge.$_{35}$, $support(C_4, KB) = \frac{1}{3}$. □

4 An Algorithm for Frequent Concept Mining

In this section we introduce an algorithm for mining frequent concepts, starting with a description of all its key components.

4.1 Canonical Form for $\mathcal{EL}^{++}$ Concepts

During traversing the space of possible patterns (concepts), one may encounter many syntactically different concept descriptions that are semantically equivalent. Therefore it is convenient to define a canonical form, to which all equivalent concepts can be transformed.

In order to define an $\mathcal{EL}^{++}$ canonical form, we firstly introduce some notation. Let by prim(C) denote the set of all the primitive concepts that occur at the top-level conjunction of C, by abst(C), and conc(C) the sets of abstract and concrete roles respectively, and by ex$_R(C)$ denote the set of the concept descriptions C' that occur in existential restrictions $\exists R_i.C'$ at the top-level conjunction of C. Let us further by $\leq_{\text{prim}}$ denote a relation that totally orders all primitive classes. The above order reflects an order of concepts in the concept subsumption hierarchy (obtained by a DL reasoner after classification) searched depth-first. By $\leq_{\text{nom}}$ let us denote a relation that totally orders the set of all individuals. The relation $\leq_{\text{nom}}$ employs ordering on classes and adds another level of ordering which orders members of particular classes within these classes (e.g. by lexicographical order). Finally, let us by $\leq_R$ denote a relation that totally orders all abstract roles in such a way, that it reflects an order of the roles in the role hierarchy (obtained by a DL reasoner after classification) searched depth-first, and by $\leq_P$ denote a relation that totally orders all concrete roles in a way reflecting an order of the roles in the concrete role hierarchy.

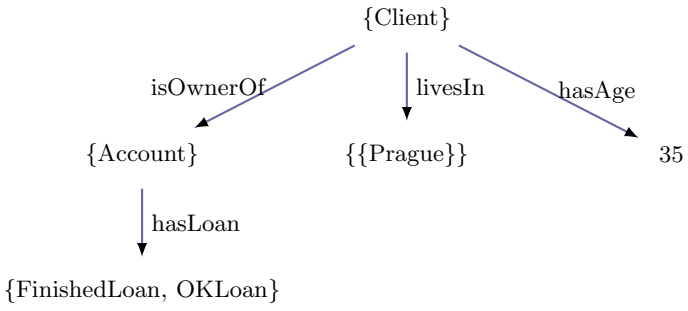


Fig. 1. A description tree for the concept C_4 from Example 2

In all the cases, only the first occurence of each term is taken into account during ordering (in case of multiple inheritance). The canonical form can then be defined as follows:

Definition 3 ($\mathcal{EL}^{++}$ canonical form). *A concept description C is in $\mathcal{EL}^{++}$ canonical form iff $C \equiv \top$ or $C \equiv \bot$ or $C \equiv \{a\}$ or if*

$$C = \prod_{\substack{P_i \in \mathsf{prim}(C) \\ i=1\ldots n}} P_i \sqcap \prod_{\substack{R_k \in N_R \\ C' \in \mathsf{ex}_R(C) \\ k=1\ldots s}} \exists R_k.C' \sqcap \prod_{\substack{P_l \in N_P \\ l=1\ldots t}} \exists P_l.f$$

where the set $\mathsf{prim}(C)$ is totally ordered by the relation $\leq_{\mathsf{prim}}$, the set of all abstract roles $R_k \in N_R, k = 1\ldots s$ is totally ordered by the relation $\leq_R$, the set of all concrete roles $P_l \in N_P, l = 1\ldots t$ is totally ordered by the relation $\leq_P$, and for any $R_k \in N_R$, every concept description C' in $\mathsf{ex}_R(C)$ is in canonical form. In general, each of the higher level intersection may be also replaced by $\top$.

$\mathcal{EL}^{++}$ concepts can be viewed as directed labeled trees, similarly as described in [13,14]. The $\mathcal{EL}^{++}$ *concept tree* $T = (V, E)$ is a directed labeled tree, where V is the finite set of *nodes*, and $E \subseteq V \times N_{R|P} \times V$ is the set of *edges*. The root of the tree is labelled with either $\top$, $\bot$ or all concept names occuring in $\mathsf{prim}(C)$. For each existential restriction $\exists R_k.C'$ occuring at the top level of C, it has an R_k-labelled edge to the root of a subtree corresponding to C'. If C' is of the form $\{a\}$, then the subtree is just a leaf, as the nodes corresponding to nominal concepts are not expanded further. For each existential restriction $\exists P_l.f$ occuring at the top level of C, it has an P_l-labelled edge to a leaf corresponding to value f (the nodes of the tree that correspond to concrete values are neither further expanded). An empty label in a node is equivalent to the top concept ($\top$). Figure 1 shows a description tree for the concept C_4 from Example 2.

4.2 Refinement Operator

Below, we define a refinement operator that constructs concepts in the form of $\mathcal{EL}^{++}$ concept descriptions.

Definition 4 (Downward refinement operator ρ). *Given is an $\mathcal{EL}^{++}$ concept description C in a canonical form (as in Definition 3). For clarity, let us introduce the following notation: by* PRIM *denote the conjunction of primitive concepts at the highest level of C, by* ABST *denote the conjunction of existential restrictions involving abstract roles, and by* CONC *the conjunction involving concrete roles, at the highest level of C respectively. The downward refinement operator ρ is a function that returns a set of the refined concepts $C' \in \rho(C)$. ρ is defined by the following operations (named according to respective manipulations on a tree T reflecting concept description C):*

(a) extend node label by primitive concept*:*

$$C' \in \rho(C) \text{ if } C' = \bigsqcap_{\substack{P_i \in \mathsf{prim}(C) \\ i=1\ldots n}} P_i \sqcap P_{n+1} \sqcap \mathsf{ABST} \sqcap \mathsf{CONC}$$

where P_{n+1} is a primitive concept, and $P_n \leq_{\mathsf{prim}} P_{n+1}$

(b) refine node label by primitive concept*:*

$$C' \in \rho(C) \text{ if } C' = \bigsqcap_{\substack{P_i \in \mathsf{prim}(C) \\ i=1\ldots j-1}} P_i \sqcap P_{j'} \sqcap \bigsqcap_{\substack{P_i \in \mathsf{prim}(C) \\ i=j+1\ldots n}} P_i \sqcap \mathsf{NOM} \sqcap \mathsf{ABST} \sqcap \mathsf{CONC}$$

where $P_{j'}$ is a primitive concept, $KB \models P_{j'} \sqsubseteq P_j$, $P_j \in \mathsf{prim}(C)$, and $P_{j-1} \leq_{\mathsf{prim}} P_{j'} \leq_{\mathsf{prim}} P_{j+1}$,

(c) append edge representing abstract role*:*

$$C' \in \rho(C) \text{ if } C' = \mathsf{PRIM} \sqcap \bigsqcap_{\substack{R_k \in N_R \\ C'' \in \mathsf{ex}_R(C) \\ k=1\ldots s}} \exists R_k.C'' \sqcap \exists R_{s+1}.\top \sqcap \mathsf{CONC}$$

where $R_s \leq_R R_{s+1}$,

(d) append edge representing abstract role with nominal filler*:*

$$C' \in \rho(C) \text{ if } C' = \mathsf{PRIM} \sqcap \bigsqcap_{\substack{R_k \in N_R \\ C'' \in \mathsf{ex}_R(C) \\ k=1\ldots s}} \exists R_k.C'' \sqcap \exists R_{s+1}.\{a\} \sqcap \mathsf{CONC}$$

where $R_s =_R R_{s+1}$,

(e) refine edge representing abstract role*:*

$$C' \in \rho(C) \text{ if } C' = \mathsf{PRIM} \sqcap \bigsqcap_{\substack{R_k \in N_R \\ C'' \in \mathsf{ex}_R(C) \\ k=1\ldots j-1}} \exists R_k.C'' \sqcap \exists R_{j'}.C'' \sqcap \bigsqcap_{\substack{R_k \in N_R \\ C'' \in \mathsf{ex}_R(C) \\ k=j+1\ldots s}} \exists R_k.C'' \sqcap \mathsf{CONC}$$

where $R_{j'} \sqsubseteq R_j$, $R_j \in \mathsf{abst}(C)$, and $R_{j-1} \leq_R R_{j'} \leq_R R_{j+1}$.

(f) append edge representing concrete role*:*

$$C' \in \rho(C) \text{ if } C' = \mathsf{PRIM} \sqcap \mathsf{ABST} \sqcap \prod_{\substack{P_l \in N_P \\ l=1\ldots t}} \exists P_l.f \sqcap \exists P_{t+1}.f'$$

where $P_t \leq_P P_{t+1}$*,*

(g) refine edge representing concrete role*:*

$$C' \in \rho(C) \text{ if } C' = \mathsf{PRIM} \sqcap \mathsf{ABST} \sqcap \prod_{\substack{P_l \in N_P \\ l=1\ldots j-1}} \exists P_l.f \sqcap \exists P_{j'}.f \sqcap \prod_{\substack{P_l \in N_P \\ l=j+1\ldots t}} \exists P_l.f$$

where $P_{j'} \sqsubseteq P_j$*,* $P_j \in \mathsf{conc}(C)$*, and* $P_{j-1} \leq_P P_{j'} \leq_P P_{j+1}$

(h) recursively refine a subtree by applying refinement operator ρ*:*

$$C' \in \rho(C) \text{ if } C' = \mathsf{PRIM} \sqcap \prod_{\substack{R_k \in N_R \\ D \in \mathsf{ex}_{R_j}(C) \\ C'' \in \mathsf{ex}_R(C) \setminus \{D\} \\ k=1,\ldots,j-1,j+1,\ldots s}} \exists R_k.C'' \sqcap \exists R_j.D' \sqcap \mathsf{CONC}$$

where $D' \in \rho(D)$*.* □

The refinement operator ρ either: (a) adds new conjunct in the form of a primitive concept (b) replaces a primitive concept by its primitive subconcept, (c) adds new conjunct in the form of an existential restriction involving abstract role with $\top$ filler, (d) adds new conjunct in the form of an existential restriction involving abstract role with a nominal filler, (e) replaces an abstract role by its subrole, (f) adds new conjunct in the form of an existential restriction involving concrete role with a filler, (g) replaces a concrete role by its subrole, or (h) recursively refines a filler of an abstract role by an application of ρ. All of these operations take the total orderings of terms into account, and thus only refinements in the canonical form are produced.

4.3 Searching Pattern Space

A high-level algorithm for mining patterns is shown in Alg. 1. It works level-wise: it repeatedly generates candidate concepts c_i using the refinement operator ρ, and tests their frequency. Only top-k frequent patterns are used to generate subsequent candidates at each iteration (best first-search is perfomed).

The usage of an expressive pattern language, and the presence of Open World Assumption (leading to less constraints on possible patterns), may result in a large pattern search-space. Thus, further steps are necessary to prune the space explored by the operator. This is usually done in ILP by introducing *declarative bias*. To this end, we have implemented a declarative bias $\mathcal{B}$ enabling to specify: i) terms to be employed for constructing refinements (concepts, roles), ii) individuals to be employed in refinements involving an abstract role with a nominal

```
Algorithm Fr-ONT
input  : C_ref, k, KB, B, MAXLEVEL
output: F
F_1 ← C_ref; C_1 ← ∅; l ← 1;
while l < MAXLEVEL do
    C_{l+1} ← ∅;
    i ← 0;
    while i < k and i < size(F_l) do
        C_{l+1} ← C_{l+1} ∪ ρ(f_i ∈ F_l);
        i ← i + 1;
    end
    foreach c_i ∈ C_{l+1} do
        if c_i is satisfiable w.r.t. KB then
            evaluate(c_i);
            if c_i is frequent then
                F_{l+1} ← c_i;
            end
        end
    end
    F_{l+1} ← sort_descending(F_{l+1});
    l ← l + 1;
end
F ← F_1 ∪ ... ∪ F_MAXLEVEL
```

Algorithm 1. The main routine of the Fr-ONT algorithm

concept filler, iii) applicable fillers for a particular concrete role.

To ensure that only concepts in the canonical form are generated, data structures are maintained that store term ordering information filtered by $\mathcal{B}$, and pointers are kept to a current term taken from a given data structure as refinement. Only the next terms in order are considered to be added as the pattern's further refinements. We also exploit taxonomical knowledge during searching pattern space, to leverage the anti-monotonicity property of support.

4.4 Implementation

The proposed algorithm has been implemented in Java. The implementation uses Pellet[2] reasoning engine with Jena API.

Below we present the results of running Fr-ONT on a FINANCIAL dataset annotated by an ontology. The dataset, already referenced in this paper, contains information from banking domain such as client demographic data, accounts, credit cards, etc. Fr-ONT has been run on a benchmark ontology built basing on this dataset, and on the part of the data concerning gold credit card holders[3]. Below we present sample patterns generated for *minsup*=0.2,

[2] http://clarkparsia.com/pellet/

[3] http://www.cs.put.poznan.pl/alawrynowicz/goldDLP2.owl

maxlevel=9, k=50, and a declarative bias: $\hat{C}$= Client, primitive concepts = {Client, Man, Woman, Account, AgeValue, Region, Loan, FinishedLoan, ProblemLoan, OKLoan, RunningLoan}, abstract roles = {isOwnerOf, livesIn, hasAgeValue, hasStatementIssuanceFrequency, hasLoan}, and where for the roles livesIn, and hasStatementIssuanceFrequency a list of all instances of Region, and StatementIssuanceFrequency (such as rPrague or sivfMonthly respectively) has been indicated as possible fillers. An example pattern obtained at the level 3 is Client ⊓∃isOwnerOf.Account ⊓∃livesIn.{rNorthMoravia} (*supp*=0.20), and denotes "*a client living in North Moravia, who owns an account*", and at the level 5 is Client ⊓∃isOwnerOf.(Account ⊓∃hasStatementIssuanceFrequency.{sifvMonthly}) ⊓ ∃hasAgeValue.{avFrom50To65} (*supp*=0.33), and denotes "*a client at age between 50 and 65, who owns an account from which statements are issued monthly*". For the given parameters, Fr-ONT stopped generating patterns at level 8, where some sample pattern is Woman ⊓ ∃isOwnerOf.(Account ⊓∃ hasStatementIssuanceFrequency.{sifvMonthly}) ⊓ ∃livesIn.Region ⊓∃hasAgeValue.AgeValue (*supp*=0.33).

Subsequently, a sample result of a run on the same sets of selected terms, with $\hat{C}$=Account, and *minsup*=0.1, at the level 4, where Fr-ONT stopped, is Account ⊓ ∃hasStatementIssuanceFrequency.⊤ ⊓ ∃hasLoan.ProblemLoan (*supp*=0.10) (interestingly any corresponding pattern with OKLoan has not been produced).

Below we also present the sample patterns generated by running Fr-ONT on another benchmark ontology, SWRC, representing knowledge about researchers and research communities, and on an associated testbed[4], for *minsup*=0.05, maxlevel=6, k=30, and a declarative bias: $\hat{C}$ = Publication, primitive concepts = {Person, Employee, AcademicStaff, FacultyMember, Lecturer, Student, Graduate, PhDStudent, Publication, Article, Book, Booklet, InCollection, InProceedings}, abstract roles = {author, editor}, concrete roles = {year}, the list of possible fillers for year = {2002, 2003, 2004, 2005}. The listed primitives terms are in taxonomic relationships in SWRC, such as for example: InProceedings ⊑ Publication, Student ⊑ Person, Graduate ⊑ Student, PhDStudent ⊑ Graduate, Employee ⊑ Person, AcademicStaff ⊑ Employee. Sample patterns at the 6th level, where Fr-ONT exploited these taxonomic hierarchies and/or employed the concrete role are as follows: Publication ⊓ ∃author.(AcademicStaff ⊓ Student) ⊓ $\exists \text{year}._{2003}$ (*supp*=0.06), InProceedings ⊓ ∃author.PhDStudent (*supp*=0.24).

5 Conclusions and Future Work

To the best of our knowledge, this is the first proposal for an algorithm for mining frequent patterns, expressed as concepts represented in description logics. Some basic ideas for this research were initially introduced in a position paper [15].

The primary motivation of this work is a future application of the proposed method in real-life scenarios. Our algorithm has been designed having some of those in mind. One such scenario, we plan an extensive experimental evaluation for, is the task of meta-mining exploiting background knowledge on data mining

[4] http://km.aifb.uni-karlsruhe.de/ws/eon2006/ontoeval.zip

domain represented in ontologies. Another scenario we consider is mining frequent patterns in the context of recommender systems using ontologies on multimedia resources. We will also investigate a suitable declarative bias language for frequent concept mining, and optimization techniques for the implementation.

Acknowledgements. This work is supported by the European Community 7th framework ICT-2007.4.4 (No 231519) "e-LICO: An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science".

References

1. Nienhuys-Cheng, S., de Wolf, R.: Foundations of Inductive Logic Programming. LNCS (LNAI), vol. 1228. Springer, Heidelberg (1997)
2. Dehaspe, L., Toivonen, H.: Discovery of frequent Datalog patterns. Data Mining and Knowledge Discovery 3(1), 7–36 (1999)
3. Nijssen, S., Kok, J.: Faster association rules for multiple relations. In: Proc. of the 17th Int. Joint Conference on Artificial Intelligence (IJCAI 2001), pp. 891–897 (2001)
4. de Raedt, L., Ramon, J.: Condensed representations for inductive logic programming. In: Proc. of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR 2004), pp. 438–446 (2004)
5. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2003)
6. Lisi, F., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning Journal 55(2), 175–210 (2004)
7. Józefowska, J., Ławrynowicz, A., Łukaszewski, T.: The role of semantics in mining frequent patterns from knowledge bases in description logics with rules. Theory and Practice of Logic Programming 10(3), 251–289 (2010)
8. Berka, P.: Guide to the financial data set. In: PKDD 2000 Discovery Challenge (2000)
9. Kietz, J.U., Morik, K.: A polynomial approach to the constructive induction of structural knowledge. Machine Learning 14(2), 193–218 (1994)
10. Iannone, L., Palmisano, I., Fanizzi, N.: An algorithm based on counterfactuals for concept learning in the Semantic Web. Appl. Intell. 26(2), 139–159 (2007)
11. Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL concept learning in description logics. In: Železný, F., Lavrač, N. (eds.) ILP 2008. LNCS (LNAI), vol. 5194, pp. 107–121. Springer, Heidelberg (2008)
12. Lehmann, J.: DL-learner: Learning concepts in description logics. Journal of Machine Learning Research (JMLR) 10, 2639–2642 (2009)
13. Baader, F., Molitor, R., Tobies, S.: Tractable and decidable fragments of conceptual graphs. In: Tepfenhart, W.M., Cyre, W.R. (eds.) ICCS 1999. LNCS, vol. 1640, pp. 480–493. Springer, Heidelberg (1999)
14. Lehmann, J., Haase, C.: Ideal downward refinement in the $\{EL\}$ description logic. In: De Raedt, L. (ed.) ILP 2009. LNCS, vol. 5989, pp. 73–87. Springer, Heidelberg (2010)
15. Ławrynowicz, A.: Foundations of frequent concept mining with formal ontologies. In: Proc. of the ECML/PKDD 2010 Workshop on Third Generation Data Mining: Towards Service-oriented Knowledge Discovery (SoKD-10), pp. 45–50 (2010)

Evaluation of Feature Combination Approaches for Text Categorisation

Robert Neumayer and Kjetil Nørvåg

Department of Computer and Information Science,
Norwegian University of Science and Technology, Trondheim, Norway

Abstract. Text categorisation relies heavily on feature selection. Both the possible reduction in dimensionality as well as improvements in classification performance are highly desirable. To the end of feature selection for text, a range of different methods have been developed, each having unique properties and selecting different features. However, it remains unclear which of them can be combined and what benefits this brings with it. In this paper we present correlation methods for the analysis of feature rankings and evaluate the combination of features according to these metrics. We further show results of an extensive study of feature selection approaches using a wide range of combination methods. We performed experiments on 19 test collections and report our findings.

1 Introduction

The automatic assignment of text documents in predefined categories is denoted to as text categorisation (TC) and has been subject to intense research for decades. However, driven by the ongoing growth of online sources and widespread availability of text documents of all kinds in electronic form, text categorisation has not lost attraction as a research area. Besides, a lot of research output has successfully been turned into applications by industry or is followed up in other research projects, e.g. News or e-mail articles are automatically sorted and delivered to end users. Spam detection techniques have reached a high level of accuracy and in many cases keep inboxes useful. Together with the growth of user generated content on the Web, this generates a strong strong demand for highly effective solutions to the text categorisation problem.

Using all the terms in the collection as feature set leads to the problem of high-dimensionality shared with all other research areas dealing with text. This dimensionality is often prohibitively high for many learning algorithms which are later used to decide which category a document is assigned to. For this reason it is required to limit the space complexity of the text categorisation problem. Feature selection is vital to facilitate such a reduction in dimensionality and most machine learning algorithms could not be applied at all without it.

A range of methods have been suggested and evaluated to this end – with varying performance. Computational resources have become easier available and make the computation of multiple feature rankings possible or applicable. For this reason it has become feasible to use more than one feature selection technique and combine their impact on classification performance. However, not

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 438–448, 2011.

enough research has been done on the possible benefits of combining the results of more than one method. Some methods are more promising to combine than others, which ones to choose from is one of the central questions we try to resolve.

The main four contributions of this paper are: a) we compare a range of feature selection and introduce new ranking merging methods which we compare to existing work; b) we further examine ways of combining them and provide a thorough analysis of which methods to combine based on both the correlations of individual rankings and performance considerations; c) additionally, we document possible performance increases and provide hints as to when the different combination methods work best; d) we show improvements by means of feature ranking merging in an extensive study based on 19 different text test collections, focusing on possible generalisations of our findings.

We continue with giving an overview of related work in the area of text categorisation in Sec. 2. This is followed by an overview of the 15 feature selection methods to be used in Sec. 3. After that, we give an analysis of the combination of these methods in Sec. 4. In Sec. 4 we provide several combination methods based on both ranks and individual values. We further describe experimental results in Sec. 5. Then, we conclude and give an outlook on future work.

2 Related Work

A good overview and a comprehensive survey of the whole area of text categorisation is given by Sebastiani in [11]. Feature selection for text categorisation is surveyed in [12]. An comparison of feature selection using linear classifier weights is given in [5]. Unsupervised feature selection has been used in the context of P2P systems in [7]. The results of a more recent and extensive empirical study of a wide range of single feature selection measures is presented in [2]. Here, the author compares a list of 11 feature selection methods. The evaluation is done on 19 test collections of different size and difficulty. The author uses one-against-all classification and as such averages all results over 229 binary classification problems. A possible combination of methods is not considered.

Social choice voting models have successfully been applied to improve meta search in information retrieval in [6]. The authors show that the Condorcet ranking merging method outperforms the Borda method with respect to precision achieved on the TREC collection. Recent research shows the superiority of reciprocal rank merging for the information retrieval problem of similarity ranking merging. This is shown by several TREC experiments in [1].

Combination experiments for text categorisation are reported in [9]. Experiments are done with four different feature selection methods and a test collection sampled from RCV1-v2. It is shown that certain combination methods improve peak R-precision and F_1. Feature selection combination was, for example, suggested in [10]. The authors selected feature selection methods based on 'uncorrelatedness' and presented results for two document collections. Both studies only partly work with benchmark collections and the results are therefore difficult to compare also due to the impact of different preprocessing techniques applied.

Table 1. Notation for feature selection

Variable	Explanation
N	total #documents in the collection.
N_{C_k}	#documents in category C_k.
$N_{\overline{C_k}}$	#documents not in category C_k.
N_F	#documents containing feature F.
$N_{\overline{F}}$	#documents not containing feature F.
N_{F,C_k}	#documents containing feature F in category C_k.
$N_{\overline{F},C_k}$	#documents not containing feature F in category C_k.
$N_{F,\overline{C_k}}$	#documents containing feature F not in category C_k.
$N_{\overline{F},\overline{C_k}}$	#documents not containing feature F not in category C_k.

Initial results of an evaluation study on a large set of categorisation problems were presented in [8]. However, in this paper we present a more in-depth analysis of feature correlation and provide experimental results for this analysis.

3 Feature Selection Methods

We list the used notation and the different feature selection methods we use in this paper in Tab. 1. We chose to use a generalised notation since we consider it easier to follow compared to the wide range of different notations. The individual feature selection methods are given by name and abbreviation in Tab. 2. A method is called unsupervised if it does not rely on previously assigned labels (methods belonging there are shown in the first part of the table). A method is called supervised if it does rely on previously assigned labels to compute the discriminative power of a feature (shown in the second part of the table). This represents a good overview of methods used in various recent studies.

4 Combination of Feature Selection Methods

The question of which feature selection techniques to combine is the most important decision. We present important considerations in the following.

4.1 Compatibility between Methods

The range of values provided by the different methods might be inhibitive in their combination based on feature value. This becomes apparent in Fig. 1. We present the value distribution of the top 100 measure for two selected methods in the training set of the 20newsgroups collection and the distribution of a random ranking for the purpose of comparison (random numbers are generated in experiments). The values are normalised between zero and one. Nevertheless the distribution is important since only measures with similar distributions can be combined in a straight-forward way. In the worst case this will lead to single techniques having little or even no effect on the final ranking (e.g. when

Table 2. Notation, unsupervised, and supervised feature selection methods used

Method	Explanation
Document Freq. (DF)	The number of documents a term occurs in
Inverse Document Freq. (IDF)	$\frac{N}{DF(F)}$
Collection Freq. (CF)	The number of occurrences of a term in a collection
Inverse Collection Freq. (ICF)	$CF(F)\log_2 \frac{N}{DF(F)}$
Term Freq. Doc. Freq. (TFD)	$(n_1 \times n_2 + c(n_1 \times n_3 + n_2 \times n_3))$, where c is a constant, $c \geq 1$, n_1 is the number of documents without the feature, n_2 is the number of documents where the feature occurs exactly once, n_3 is the number of documents where the feature occurs twice or more.
Information Gain (IG)	$-\sum_{k=1}^{C} \frac{N_{C_k}}{N} \ln \frac{N_{C_k}}{N} + \frac{N_F}{N} \sum_{k=1}^{C} \frac{N_{F,C_k}}{N_F} \ln \frac{N_{F,C_k}}{N_F}$ $+\frac{N_{\overline{F}}}{N} \sum_{k=1}^{C} \frac{N_{\overline{F},C_k}}{N_{\overline{F}}} \ln \frac{N_{\overline{F},C_k}}{N_{\overline{F}}}$
Mutual Information (MI)	$\sum_{v_f \epsilon \{1,0\}} \sum_{v_{C_k} \epsilon \{1,0\}} P(F = v_f, C_k = v_{C_k})$ $\ln \frac{P(F=v_f, C_k=v_{C_k})}{P(F=v_f)P(C_k=v_{C_k})}$
Odds Ratio (OR)	$\ln \frac{P(F\|C_k)(1-P(F\|\overline{C_k}))}{P(F\|\overline{C_k})(1-P(F\|C_k))} = \ln \frac{\left(\frac{N_{F,C_k}}{N_{C_k}}\right)\left(1-\frac{N_{F,\overline{C_k}}}{N_{\overline{C_k}}}\right)}{\left(\frac{N_{F,\overline{C_k}}}{N_{\overline{C_k}}}\right)\left(1-\frac{N_{F,C_k}}{N_{C_k}}\right)}$
Class Discrimination Meas. (CDM)	$\sum_{k=1}^{\|C\|} \left\| \log \frac{\left(\frac{N_{F,C_k}}{N_{C_k}}\right)}{\left(\frac{N_{F,\overline{C_k}}}{N_{\overline{C_k}}}\right)} \right\|$
Word Freq. (WF)	N_{F,C_k}
χ^2 statistic (χ^2)	$\frac{N \times \left((N_{F,C_k} \times N_{\overline{F},\overline{C_k}}) - (N_{F,\overline{C_k}} \times N_{\overline{F},C_k})\right)^2}{N_F \times N_{\overline{F}} \times N_{C_k} \times N_{\overline{C_k}}}$
NGL-Coefficient (NGL)	$\frac{\sqrt{N}(N_{F,C_k} N_{\overline{F},\overline{C_k}} - N_{F,\overline{C_k}} N_{\overline{F},C_k})}{\sqrt{N_F N_{\overline{F}} N_{C_k} N_{\overline{C_k}}}}$
Categorical Proportional Difference (CPD)	$\frac{N_{F,C_k} - N_{F,\overline{C_k}}}{N_F}$
GSS-Coefficient	$N_{F,C_k} N_{\overline{F},\overline{C_k}} - N_{F,\overline{C_k}} N_{\overline{F},C_k}$
Bi-Normal Separation (BNS)	$\left\| F^{-1}\left(\frac{N_{F,C_k}}{N_{C_k}}\right) - F^{-1}\left(\frac{N_{F,\overline{C_k}}}{N_{\overline{C_k}}}\right) \right\|$

one method provides consistently higher values than the other). These findings should be taken into account when deciding what techniques to combine. We therefore stress the importance of normalisation for ranking combination.

4.2 Fitness of Individual Methods

The performance of the individual methods is definitely an important criterion when choosing which combinations to combine. We assembled a list of methods with varying individual performance and different numbers of features.

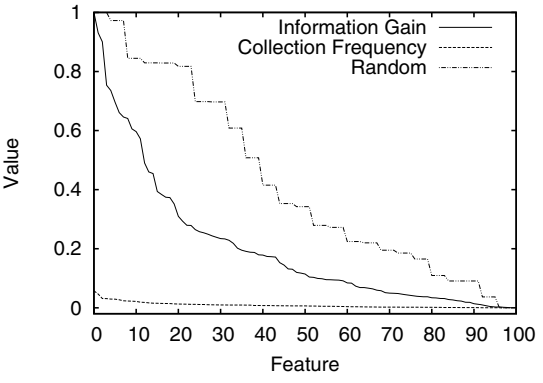


Fig. 1. Normalised feature values for top 100 features in the 20-news collection for two selected feature selection methods compared to the random distribution

4.3 Correlations of Individual Methods

In this section we analyse different possibilities of which rankings or feature selection techniques to combine. The more correlation there is between two rankings the less benefit can be expected from their combination (i.e. if two methods provide equally good results but have low correlation, it can be assumed that different features are responsible). The main goal here is to find groups of non-correlating rankings produced by different feature selection methods. We use the Spearman rank coefficient to find pairwise correlations between rankings. It is a measure for correlation between two rankings, operating on the rank on their elements rather than their numeric values; the coefficient is given in the following (d denotes the difference in ranks and n the length of the rankings, i.e. the number of features selected per ranking).

$$R = 1 - \frac{6 \sum d^2}{n^3 - n}$$

4.4 Actual Combination Techniques

A range of methods have been suggested for ranking, albeit often in different settings like Condorcet merging and Borda merging which initially are originally used for defining winners of elections.

Two or several rankings can be combined by using the ranks of the terms in the individual rankings. When dealing with two rankings of a term t_i, the ranks of this term $r_j(t_i)$ are used rather than the plain values. If term t_x is ranked first in r_1 and second in ranking r_2, these rank values are $r_1(t_x) = 1$ and $r_2(t_x) = 2$ respectively. If ranks from several methods are combined the final list is sorted according to the newly computed rank values.

Another possibility is to use the values given by the individual methods. Term t_x might for example have different values in different rankings. To get a final value for a term across multiple rankings these individual values might be combined by, e.g. building the sum or average over the values. These final values are then sorted and the top k features selected as input to the classifier.

Table 3. Ranking Merging methods used

Method	Explanation
Highest Rank (HR)	A feature's highest rank in all single rankings.
Lowest Rank (LR)	The lowest of all rankings is used as final score.
Average Rank (AR)	The average over all single ranks is used.
Borda Ranking Merging (BRM)	Gives scores according to the length of the single rankings. If the size of a ranking is n and an element is ranked at the ith position the score $\frac{i}{n}$. This technique is also applicable for individual rankings with different lengths. The final scores are the sum of the individual scores.
Condorcet Ranking Merging (CRM)	A majoritarian method favouring the candidate beating every other candidate in pair-wise comparisons. If, e.g., feature a is higher ranked than b in any of the methods, it a clearly beats b. For aggregation the number of pair-wise wins or ties is summed for each candidate and the one with the highest score is the overall winner.
Reciprocal Ranking Merging (RRM)	In this setting, the score for a feature is the sum of 1 divided by the rank in the single rankings.
Divide by Max. then OR (DMOR)	The average over all single feature values in this setting we normalise by the maximum.
Divide by Length then OR (DLOR)	Normalisation by the length of the vector.
Pure Round Robin (RR)	One feature from each ranking is added to the final ranking in turn until the desired number of features is reached.
Top N Ranking Merging (Top$N$)	The top n features from each ranking in turn are added until enough features are collected.
Weighted N Ranking Merging (WN)	The first n % are taken from the first ranking, the remaining $1 - n$ % are composed of the other rankings in equal parts.

We introduce a third group of methods based on round robin algorithms and weighted combinations. The rationale behind the weighted methods is that the whole set of features selected by one method is more than the sum of its parts. This means that it's well possible that the performance of a method is influenced not by the single features but that there is an underlying dependence on the features. With weighted ranking merging the majority of the features is selected from one method and only a smaller fraction from additional methods.

5 Experiments

The following experiments were performed using the 20newsgroups data set[1], which has become very popular for text experiments in the field of machine

[1] http://people.csail.mit.edu/jrennie/20Newsgroups

Table 4. Spearman rank coefficient measure for the full set of documents for each feature selection method. We list statistics of correlation values for each method. The evaluation considers all features in the training set of the 20news collection.

	IG	OR	WF	MI	CS	DIA	NGL	CPD	BNS	CDM	GSS	DF	TFD	W	CF	ICFI
IG	1.0	.33	.91	.95	.81	**-.28**	**.10**	**-.32**	**-.03**	**-.10**	.90	.92	.80	.21	.88	.86
OR	.33	1.0	.41	.38	.60	**.12**	**.14**	.51	**-.03**	.52	.53	**.10**	**.14**	**.14**	**.13**	**.13**
WF	.91	.41	1.0	.98	.85	**-.00**	.27	**-.16**	**.02**	**-.03**	.92	.83	.75	.24	.81	.79
MI	.95	.38	.98	1.0	.84	**-.07**	.23	**-.22**	**-.02**	**-.05**	.91	.88	.78	.24	.85	.83
CS	.81	.60	.85	.84	1.0	**-.05**	.19	.17	**.02**	.31	.95	.59	.57	.30	.60	.59
DIA	**-.28**	**.12**	**-.00**	**-.07**	**-.05**	1.0	.80	.44	**-.10**	.34	**-.15**	**-.28**	**-.22**	.18	**-.25**	**-.25**
NGL	**.10**	**.14**	.27	.23	.19	.80	1.0	.20	**-.12**	.22	**.14**	**.09**	**.10**	.29	**.10**	**.10**
CPD	**-.32**	.51	**-.16**	**-.22**	.17	.44	.20	1.0	**-.23**	.85	**-.03**	**-.52**	**-.43**	.17	**-.48**	**-.47**
BNS	**-.03**	**-.03**	**.02**	**-.02**	**.02**	**-.10**	**-.12**	**-.23**	1.0	**-.38**	**.08**	**.01**	**.09**	**-.11**	**.05**	**.06**
CDM	**-.10**	.52	**-.03**	**-.05**	.31	.34	.22	.85	**-.38**	1.0	**.11**	**-.36**	**-.30**	.26	**-.32**	**-.32**
GSS	.90	.53	.92	.91	.95	**-.15**	**.14**	**-.03**	**.08**	**.11**	1.0	.74	.69	.24	.73	.72
DF	.92	**.10**	.83	.88	.59	**-.28**	**.09**	**-.52**	**.01**	**-.36**	.74	1.0	.84	.16	.93	.90
TFD	.80	**.14**	.75	.78	.57	**-.22**	**.10**	**-.43**	**.09**	**-.30**	.69	.84	1.0	.16	.96	.97
W	.21	**.14**	.24	.24	.30	.18	.29	.17	**-.11**	.26	.24	.16	.16	1.0	.17	.16
CF	.88	**.13**	.81	.85	.60	**-.25**	**.10**	**-.48**	**.05**	**-.32**	.73	.93	.96	.17	1.0	1.0
ICF	.86	**.13**	.79	.83	.59	**-.25**	**.10**	**-.47**	**.06**	**-.32**	.72	.90	.97	.16	1.0	1.0

Table 5. Overlap within feature rankings for the 20news collection. 1000 features are selected and we count the number of features occurring in both rankings.

	IG	OR	WF	MI	CS	DIA	NGL	CPD	BNS	CDM	GSS	DF	TFD	W	CF	ICF
IG	1.0	.31	.76	.95	.69	**.01**	.35	**.00**	.60	**.09**	.86	.51	.59	**.07**	.55	.56
OR	.31	1.0	.26	.31	.34	**.08**	**.21**	**.01**	**.25**	.63	.27	.29	.28	**.13**	.28	.27
WF	.76	.26	1.0	.74	.56	**.00**	.31	**.00**	.41	**.06**	.85	.70	.76	**.05**	.74	.75
MI	.95	.31	.74	1.0	.74	**.01**	.38	**.00**	.64	**.12**	.85	.47	.55	**.08**	.52	.53
CS	.69	.34	.56	.74	1.0	**.02**	.44	**.00**	.66	**.23**	.67	.28	.36	**.10**	.33	.34
DIA	**.01**	**.08**	**.00**	**.01**	**.02**	1.0	**.04**	**.11**	**.01**	**.12**	**.01**	**.00**	**.00**	**.04**	**.00**	**.00**
NGL	.35	**.21**	.31	.38	.44	**.04**	1.0	**.00**	.40	**.15**	.35	**.19**	**.23**	**.10**	**.22**	**.23**
CPD	**.00**	**.01**	**.00**	**.00**	**.00**	**.11**	**.00**	1.0	**.00**	**.02**	**.00**	**.00**	**.00**	**.02**	**.00**	**.00**
BNS	.60	**.25**	.41	.64	.66	**.01**	.40	**.00**	1.0	**.15**	.52	**.16**	**.25**	**.09**	**.22**	**.24**
CDM	**.09**	.63	**.06**	**.12**	**.23**	**.12**	**.15**	**.02**	**.15**	1.0	**.09**	**.00**	**.01**	**.16**	**.02**	**.03**
GSS	.86	.27	.85	.85	.67	**.01**	.35	**.00**	.52	**.09**	1.0	.56	.63	**.06**	.60	.61
DF	.51	.29	.70	.47	.28	**.00**	**.19**	**.00**	**.16**	**.00**	.56	1.0	.87	**.02**	.89	.85
TFD	.59	.28	.76	.55	.36	**.00**	**.23**	**.00**	**.25**	**.01**	.63	.87	1.0	**.03**	.93	.92
W	**.07**	**.13**	**.05**	**.08**	**.10**	**.04**	**.10**	**.02**	**.09**	**.16**	**.06**	**.02**	**.03**	1.0	**.03**	**.04**
CF	.55	.28	.74	.52	.33	**.00**	**.22**	**.00**	**.22**	**.02**	.60	.89	.93	**.03**	1.0	.96
ICF	.56	.27	.75	.53	.34	**.00**	**.23**	**.00**	**.24**	**.03**	.61	.85	.92	**.04**	.96	1.0

learning and has been used for example in [4]. The data set consists of newsgroup postings from the 20 newsgroups. From each newsgroup, 1.000 articles from the year 1993 have been selected; after removing duplicate articles (mostly cross-postings to several newsgroups), 18.846 unique messages remain. Each text consists of the message body and in addition the 'Subject' and the 'From' header lines which we discarded before analysis. We use the predefined 'bydate' split, which is divided into training (60%) and testing (40%).

Additionally, we use a set of categorisation problems also used for binary classification experiments in [2], which were initially used by Han and Karypis and

Table 6. Experimental results on 20news, single methods in (a), combinations in (b)

(a) Classification results for the 20news collection, 1000 features, individual methods

Method	Acc.
CF	66.76
TFD	67.75
DF	64.90
ICF	67.37
WF	71.14
IG	72.65
BNS	72.03
CPD	8.44
CHI	**73.49**
CDM	42.66
DIA	8.75
GSS	71.68
MI	72.94
NGL	60.62
OR	63.83

(b) Classification results for combinations for 1000 features. We list combinations and merging methods representing an improvement over the best single method, the best values are shown in bold font

Methods and combination type	Acc.
BNS-CHI-AvgMinMaxNorm	73.54
BNS-CHI-AvgRank	**74.03**
BNS-CHI-Borda	**74.03**
BNS-CHI-Condorcet	73.59
BNS-CHI-LowestRank	73.70
BNS-CHI-Reciprocal	**74.02**
BNS-DF-MI-CHI-WF-OR-AvgMinMaxNorm	73.54
BNS-IG-Condorcet	73.74
BNS-IG-HighestRank	73.77
BNS-IG-Reciprocal	73.77
BNS-IG-RoundRobin	73.79
BNS-IG-Top100RoundRobin	73.67
BNS-IG-Top50RoundRobin	73.70
BNS-MI-AvgRank	73.61
BNS-MI-Borda	73.61
BNS-MI-Condorcet	73.55
BNS-MI-Reciprocal	73.65
BNS-WF-AvgMinMaxNorm	73.67
IG-BNS-Condorcet	73.74
IG-BNS-HighestRank	73.77
IG-BNS-Reciprocal	73.77
IG-BNS-RoundRobin	73.73

originate from TREC, OHSUMED, Reuters. The collection sizes range from 204 to 31472 documents and the number of classes varies from six to 36 classes. All collections were already preprocessed by basic stemming and stop-word removal. Unless stated otherwise we always select the 1000 best features, this can either mean 1000 per method for the single runs or 1000 features as a combination of multiple rankings. We both assume 1000 features to be a reasonable dimensionality in terms of complexity and performance and fixed this parameter to limit the number of results.

5.1 Individual Pair-Wise Correlations

Spearman. The results of the computation of the Spearman coefficient for all terms in the corpus occurring more than once, i.e. 53000 terms is given in Tab. 4. It is shown that some methods have more un-correlated methods than others. In cases of methods which have rather low performance when used exclusively like DIA or NGL this is not surprising, in other cases like BNS (used, e.g., in [3]) it suggests that the combination of the method with others might be beneficial.

Table 7. Results on the 19 text collections, single methods in (a), combinations in (b)

(a) Averaged classification results over all 19 test collections, 1000 features, individual methods

Method	Acc.
TFD	85,24
DF	84,38
CF	84,90
WF	83,77
IG	**86,45**
BNS	84,04
CPD	71,02
CHI	86,36
CDM	73,85
DIA	52,98
GSS	85,88
MI	85,97
NGL	69,64
OR	83,68

(b) Classification results for combinations for 1000 features. We list combinations which represent an improvement over the best single method, the best values are shown in bold font

Methods and combination type	Acc.
IG-BNS-AverageMinMaxNorm	86,14
IG-BNS-Condorcet	86,38
IG-BNS-DLOR	86,82
IG-BNS-Main50	86,45
IG-BNS-Main60	86,45
IG-BNS-Main70	86,45
IG-BNS-Main80	86,45
IG-BNS-Main90	86,45
IG-BNS-RoundRobin	**86,65**
IG-BNS-Top100RoundRobin	**86,69**
IG-BNS-Top300RoundRobin	**86,66**
IG-BNS-Top50RoundRobin	**86,71**
BNS-DF-MI-CHI-WF-IG-OR-Condorcet	86,31
BNS-DF-MI-CHI-WF-OR-Main50	86,88
BNS-DF-MI-CHI-WF-OR-Main60	86,87

However, this only gives an overall view of the potential of combination of the methods. The correlation of all terms in the collection can only partly help to discriminate. If we look at the correlation only at the $top - n$ terms, the results might differ. It is for example possible that two methods have a low correlation overall, but a high correlation when only the top 1000 features are considered.

Overlap Metric. The decision on which feature selection methods to compare also relies on the correlation for the respective top-n features. To this end we chose the overlap metric which simply calculates the percentage of features occurring in both rankings (the Spearman coefficient was undefined for some rankings due to a lack of co-occurring features).

Based on Tab. 5 we suggest the following combinations of methods: BNS OR WF CDM TFD. BNS has a low overlap ($< .25$) with nine other methods and therefore constitutes a good basis for combination. The other methods have reasonable overlap with each other and belong to different classes of methods (supervised/unsupervised).

We show the results for all single methods and the 20news collection in 6(a). The best method(s) in each column are printed in bold font. For the single methods we achieved the best results with the χ^2 method, the WF, IG, BNS, GSS and MI methods are not far behind (Tab. 6(a)). We then performed experiments with combinations and ranking merging, based on the analysis provided earlier. Out of the 364 experiments performed (all pairwise combinations plus the combination of all methods selected), 22 are improvements over the χ^2 method. The improvement is, however, limited with 74.03 over 73.49 with the best single

method. Reciprocal rank merging is included in four out of the five pairwise combinations and along with reciprocal rank merging is the most common method.

We show the results achieved on the collection of 19 collections in Tab. 7, the values listed are averages over all 19 results. The best single method in this context is IG with 86.36 per cent of the instances correctly classified, shortly followed by MI, GSS, and TFD. We found marginal improvements by merging shown in Tab. 7(b). The merging methods mainly relying on one method and taking in few features from the remaining methods perform more stable.

6 Conclusions and Future Work

We presented a range of methods for both feature selection and combination for text categorisation. In addition, we presented two classes of methods not previously used for categorisation methods (round robin based and weighted ranking merging). We further presented an extensive experimental evaluation of which feature selection methods to combine and performance evaluation on a diverse set of text categorisation benchmark collections. Future work will mainly deal with new feature ranking merging strategies in limited application domains and the development of strategies when to rely on which combination of methods.

Acknowledgements

We hereby express gratitude to Østein Løhre Garnes who helped with the initial implementation of some of the feature selection methods in his master's thesis.

References

1. Cormack, G.V., Clarke, C.L.A., Büttcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd ACM SIGIR, pp. 758–759 (2009)
2. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research 3, 1289–1305 (2003)
3. Forman, G.: BNS feature scaling: an improved representation over tf-idf for SVM text classification. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM), pp. 263–270 (2008)
4. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
5. Mladenić, D., Brank, J., Grobelnik, M., Milic-Frayling, N.: Feature selection using linear classifier weights: interaction with classification models. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 25-29, pp. 234–241. ACM, New York (2004)
6. Montague, M., Aslam, J.A.: Condorcet fusion for improved retrieval. In: Proceedings of the 11th ACM International Conference on Information and Knowledge Management (CIKM), pp. 538–548 (2002)
7. Neumayer, R., Doulkeridis, C., Nørvåg, K.: A hybrid approach for estimating document frequencies in unstructured P2P networks. Information Systems 36(3), 579–595 (2011)

8. Neumayer, R., Mayer, R., Nørvåg, K.: Combination of feature selection methods for text categorisation. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 763–767. Springer, Heidelberg (2011)
9. Scott Olsson, J., Oard, D.W.: Combining feature selectors for text classification. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM), pp. 798–799 (2006)
10. Rogati, M., Yang, Y.: High-performing feature selection for text classification. In: Proceedings of the 11th ACM International Conference on Information and Knowledge Management (CIKM), pp. 659–661 (2002)
11. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
12. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning (ICML), pp. 412–420 (1997)

Towards Automatic Acquisition of a Fully Sense Tagged Corpus for Persian

Bahareh Sarrafzadeh, Nikolay Yakovets, Nick Cercone, and Aijun An

Department of Computer Science and Engineering, York University, Canada
{bahar,hush,nick,aan}@cse.yorku.ca

Abstract. Sense tagged corpora play a crucial role in Natural Language Processing, particularly in Word Sense Disambiguation and Natural Language Understanding. Since semantic annotations are usually performed by humans, such corpora are limited to a handful of tagged texts and are not available for many languages with scarce resources including Persian. The shortage of efficient, reliable linguistic resources and fundamental text processing modules for Persian have been a challenge for researchers investigating this language. We employ a newly-proposed cross-lingual sense disambiguation algorithm to automatically create large sense tagged corpora. The initial evaluation of the tagged corpus indicates promising results.

1 Introduction

Word Sense Disambiguation (WSD) is the task of selecting the most appropriate meaning for a polysemous word, based on the context in which it occurs. Recent advancements in corpus linguistics technologies and the greater availability of more and more textual data encourage researchers to employ comparable and parallel corpora to address various NLP tasks.

To exploit supervised WSD approaches for applications as Machine Translation (MT) and Information Retrieval (IR), a large amount of sense-tagged examples for each sense of a word is needed. Devising an automatic method to generate such corpora thus will be of great benefit for languages with scarce resources such as Persian.

Recently we proposed a novel cross-lingual WSD approach that takes advantage of available sense disambiguation systems and linguistic resources for English to identify the word sense in a Persian document based on a comparable English document of the same topic [1]. The method was evaluated on comparable corpora that consist of a set of pairwise articles of the same topic in English and Persian. The result was promising [1].

In this paper, we aim at creating sense-tagged corpora to aid supervised and semi-supervised WSD systems. For such a purpose, we apply our newly-proposed WSD method to a parallel corpus, which contains sentence-level translations between English and Persian. To improve performance, we also extend the cross-lingual WSD approach by adding a direct sense tagging phase and enhancing the sense transfer stage of the cross-lingual method. We evaluate the accuracy of our improved approach and report the results.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 449–455, 2011.

2 Related Work

The knowledge acquisition bottleneck is pervasive across approaches to WSD. The availability of large-scale sense tagged corpora is crucial for many NLP systems. There are two branches of efforts to overcome this bottleneck. Some aim at creating manually sense tagged corpora. Tagging is performed by lexicographers. Consequently, it is expensive, limiting the size of such corpora to a handful of tagged texts. To lower the cost and increase the coverage of the tagged corpus, some developers created manually tagged corpora (e.g. Open Mind Word Expert [2]) by distributing the annotation workload among millions of web users as potential human annotators. While most manually sense tagged corpora are developed for English [3], they are not limited to this language only [4].

Automatic creation of sense tagged corpora seeks to minimize the knowledge acquisition bottleneck inherent to supervised approaches. In [5] they acquire example sentences for senses of words automatically based on the information provided in WordNet and information gathered from the Internet using existing search engines. [6] uses an aligned English-French corpus. For each English word, the classification of contexts is done based on the different translations in French for the different word senses. A problem is that different senses of polysemous words often translate to the same word in French. For such words it is impossible to acquire examples with this method [5]. [7] uses a word-aligned English-Spanish parallel corpus, and independently applies WSD heuristics for each of the languages to obtain ranked lists of senses for each word and picks the best sense for the word based on the overlaps of these lists. [8] uses a word aligned English-Italian corpus obtained from the MultiSemCor[1] and the Italian component of MultiWordNet[2] which is aligned with WordNet to automatically acquire sense tagged data, exploiting the polisemic differential between two languages.

For Persian, there is no publicly available sense-tagged corpus to use. There have been different attempts to apply supervised approaches to WSD for which a set of manually tagged words were prepared [9], [10]. However, some researchers are working to provide linguistic resources and processing units for Persian. FarsNet 1.0 [11] is a lexical ontology that relates synsets in each POS category by the set of WordNet 2.1 relations and connects Farsi synsets to English ones (in WordNet 3.0) using inter-lingual relations.

Our approach is unique in the sense that there has been no attempt to create a sense tagged corpus using an automatic or semi-automatic approach for the Persian language. Second, thanks to the availability of FarsNet, as opposed to many cross lingual approaches, we tag Persian words using sense tags in the same language instead of using either a sense inventory of another language or translations provided by a parallel corpus. Therefore, the resulted corpus can be utilized for many monolingual NLP tasks such as IR, Text Classification as well as bilingual ones including MT and Cross-Lingual tasks. In comparison with most automatic approaches which use a bilingual parallel corpus to generate

[1] http://multisemcor.itc.it
[2] http://multiwordnet.itc.it

sense tagged corpora for a target corpus, we do not sense tag both languages independently, nor do we use translation correspondences to distinguish senses. Instead, taking advantage of available mappings between synsets in WordNet and FarsNet, we utilize an existing source language (English) sense tagger which uses WordNet as a sense inventory to sense tag the target language (Persian) words. Finally, in order to improve the recall of our system, we employ a direct sense tagging method called Extended Lesk which has never been exploited to address WSD for Persian texts.

3 Creating the Sense Tagged Corpus

A direct strategy for creating a sense tagged corpus for WSD is to use parallel corpora to identify correspondences between word pairs. We employ the cross-lingual word sense tagging method described in [1] which has a high accuracy, but a relatively low recall, to tag Persian words using corresponding English tagged words in the utilized parallel corpus. We then apply a direct knowledge based algorithm to sense tag the remaining words. We replaced the comparable corpus used in [1] with a parallel corpus. Since Persian sentences in this corpus are a direct translation of the English ones in addition to improvements we made to both English tagging and the sense transfer phases, we gain better accuracy and coverage for the tagging results.

Currently available Persian-English parallel corpora are Miangah's corpus [12][3] consisting of 4,860,000 words and Tehran (TEP) corpus [13] composed of 612,086 bilingual sentences extracted from movie subtitles. TEP is a larger corpus and freely available, but the sentences are short and informal. Miangah's is smaller in size and is not available for free, but the quality of data leads to more apropos results. The texts in the corpus include a variety of text types from different categories such as art, culture, literature and science.

Several steps of preprocessing were carried out. On the English side, tokenization, lemmatization and POS tagging were performed by the English tagger. At the Farsi side, however, we used STeP-1 [14] to perform tokenization and stemming. The other challenge with Persian text processing is that there can be identical characters with different encodings observed in different resources. These are unified during this step.

We exploited a cross lingual approach [1] to tag the word senses in Persian texts. We also applied a knowledge based method directly to the Persian sentences to improve the recall. A brief description of these two methods follows.

Cross Lingual Phase: Persian WSD using Tagged English Words. This phase consists of two separate stages. First, we use an English WSD system to assign sense tags to English words. Next, we transfer these senses to corresponding Persian words. Since, by design, these two stages are distinct, different English WSD systems can be employed in the first stage. There are different factors affecting the performance of our system.

[3] Available via European Language Resource Association (ELRA)

First the more accurate the English tagger is, the more accurate the Persian sense tags will be. Supervised systems proved to offer the highest accuracy for WSD. There are many supervised WSD systems developed for English. However, as supervised systems usually perform sense disambiguation for a small set of words, using such a system limits the coverage of our method. Therefore, currently, we utilized the unsupervised application SenseRelate [15] for the English WSD stage which performs all word sense tagging using WordNet. We selected the Extended Lesk algorithm [16] which leads to the most accurate disambiguation [15]. We evaluated and corrected the wrong tags assigned by SenseRelate in order to investigate the reliability of our cross lingual approach for assigning sense tags to Persian words assuming we have a perfectly sense tagged English side. SenseRelate tags all ambiguous words in the input English sentences. Each of these sense labels corresponds to a synset in WordNet containing that word in a particular sense. We transfer these synsets from English to Persian using interlingual relations provided by FarsNet and match each WordNet synset assigned to a word in an English sentence to its corresponding synset in FarsNet.

Second, we need to match Farsi words with their counterparts on the English side. When it is possible to apply an accurate word alignment method to the language pair under examination, the creation of the sense tagged corpus from parallel corpora can be simple. However, word alignment methods hardly present a satisfactory performance, especially in corpora of real translations, where correspondences are often not one to one [17]. Therefore, we do not employ word alignment methods, since they may convey serious errors to the tagged corpus. Instead, for each matched synset in FarsNet which contains a set of Persian synonym words, we find all these words and assign the same sense as the English label to its translations in the aligned Persian sentence.

Initial evaluation indicated some words cannot be matched at the Farsi side because Farsi synsets usually do not provide full lists of synonyms. Therefore we extended the synonym set for each Persian word, using an available English-Persian dictionary, such that, for each tagged English word from an English sentence, we find all Persian translations and add them to the Farsi synset. Although these words can convey different senses of the English word, we adjust it by giving higher priority to words which are provided by the FarsNet synset. Moreover, according to the one sense per discourse heuristic [6], it is not probable to observe same Farsi words with different senses in one sentence.

Direct Phase: Applying Extended Lesk for Persian WSD. To increase the number of tagged words in our corpus, we applied a direct WSD algorithm to Persian sentences. Thanks to the availability of FarsNet, the Extended Lesk method is applicable to Persian texts as well. Although Persian WSD while working with Persian texts directly seems to be more promising, the evaluation results indicate a better performance for the Cross Lingual system [1]. Therefore, we considered only the tags with a score higher than a predefined confidence threshold. This results in gaining a higher recall while the tags remain accurate.

4 Evaluation

The tagged corpus was evaluated on 480 words which were randomly selected from various domains such as Politics, Science, Culture, Art and had an average sense count of 2.17. Seven human experts were involved in the evaluation process. In the first step, the output from SenseRelate was revised manually and the wrong tags assigned were corrected. This led to fully accurate sense tagged English sentences. After these tags were transferred and assigned to Persian words on corresponding Persian sentences, the human experts evaluated each tagged word as "the best sense assigned", "almost accurate" and "wrong sense assigned". The second option considers cases in which the assigned sense is not the best available sense for a word in a particular context, but it is very close to the correct meaning (not a wrong sense) which is influenced by the evaluation metric proposed by Resnik and Yarowsky in [18]. Evaluation results indicate an error rate of 9% for the selected Farsi words. Table 1 summarizes these results. Studying the output results revealed the content words describing the main concept of each sentence are highly probable to receive the correct sense tag.

This system demonstrates a good accuracy of 91%, but a relatively low recall of 46%. Note that the original English tagger has an average recall of 57%. This will act as an upper bound for our system's recall. The reason for a lower recall than the English tagger is that FarsNet is still at a preliminary stage of development, and does not cover all words and senses in Persian. In terms of size, it is significantly smaller (10000 synsets) than WordNet (more than 117000 synsets) and it covers roughly 9000 relations between both senses and synsets. Another problem is tagging verbs in Persian sentences. Since verbs appear in their infinitive format in FarsNet while they are inflected in a particular tense and person, a better morphological analysis of Persian verbs is required to increase the number of matches. Moreover, structural differences between the English and Persian languages usually lead to observing single English words translating to Persian phrases or compound words. Since FarsNet does not contain all these words collocations, we might tag some part of a compound word and leave the rest untagged. Since our main goal is developing a cross-lingual, yet language independent, approach to create sense tagged corpora, we have not designed Persian-specific solutions to improve the recall at this time. Having an "ideal" aligned WordNet (a lexical resource such that all the sense distinctions in one language are reflected in the other, and all words and phrases are included) would minimize this issue.

Since the senses in FarsNet are not sorted based on their frequency of usage (as opposed to WordNet), we assigned the first sense appearing in FarsNet (for each POS) to words to create a baseline system. According to the results indicated in Table 1, applying our novel approach results in a 11% improvement in the F-score[4] in comparison with this selected baseline. However, assigning the most

[4] F-Score is calculated as $2\frac{(1-ErrorRate)\cdot Recall}{1-ErrorRate+Recall}$, where $ErrorRate$ is the percentage of words that have been assigned the wrong sense.

Table 1. Evaluation Results

	Cross Lingual			Cross Lingual + Direct			Baseline		
	P	R	F-Score	P	R	F-Score	P	R	F-Score
Best Sense	80%			76%			45%		
Almost Accurate	11%	0.46	0.60	8%	0.57	0.67	11%	0.46	0.49
Wrong Sense	9%			16%			44%		

frequent sense to Persian words would be a more realistic baseline which we plan to employ once it is made available for FarsNet.

The untagged words remaining from Cross-lingual phase were sense tagged using the Direct approach. Since the final tagged corpus should be highly accurate, we did not sacrifice accuracy to gain a higher recall. Therefore, we considered a minimum score of 8[5], and approved the tags with an associate score of equal to or higher than this threshold. This results in an improvement of 11% in recall at a cost of 6% in accuracy. Due to the small size of FarsNet and the relatively higher error rate of the Direct approach, an improvement in the recall resulted in a decrease in accuracy. Hence, exploiting the Cross Lingual approach without passing the results through the Direct phase will result in obtaining a more accurate tagged corpus while the recall remains about 11% lower.

5 Conclusions and Future Work

We proposed an automatic approach for creating fully sense-tagged corpora for the Persian language which has an error rate of 9%. Although the resulted corpus might be noisy, it is still much easier and less time consuming to check already tagged data than to start tagging from scratch.

Since the accuracy of the tags assigned to the English words will affect that of Persian sense tags, a more accurate English tagger can improve the final results of our system. We are planning to replace SenseRelate with a more accurate English tagger such as WSDGate framework[6] to minimize the manual correction of English tags. Moreover, we are investigating linguistic based solutions to improve the matching desired Persian words during the Transfer phase. Finally, improvements in Word Alignment techniques For the English – Persian language pair can be of great benefit to maximize the coverage of our system.

Acknowledgements. This research is partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC). We would like to thank Prof. Shamsfard from the Natural Language Processing Research laboratory of Shahid Beheshti University (SBU) for providing us with the FarsNet 1.0 package.

[5] This threshold is set based on experiments favouring precision over recall.
[6] http://wsdgate.sourceforge.net/

References

1. Sarrafzadeh, B., Yakovets, N., Cercone, N., An, A.: Cross lingual word sense disambiguation for languages with scarce resources. In: Proc. of The 24th Canadian Conference on Artificial Intelligence (2011)
2. Chklovski, T., Mihalcea, R.: Building a sense tagged corpus with open mind word expert. In: Proc. of the ACL 2002 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, vol. 8 (2002)
3. Miller, G.A., et.al: A semantic concordance. In: Proc. of the Workshop on Human Language Technology (1993)
4. Koeva, S., Lesseva, S., Todorova, M.: Bulgarian sense tagged corpus. In: Proc. of the Fifth International Conference Formal Approaches to South Slavic and Balkan Languages (2006)
5. Mihalcea, R., Moldovan, D.I.: An automatic method for generating sense tagged corpora. In: Proc. of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference (1999)
6. Gale, W.A., Church, K.W., Yarowsky, D.: One sense per discourse. In: Proc. of the Workshop on Speech and Natural Language (1992)
7. de Melo, G., Weikum, G.: Extracting sense-disambiguated example sentences from parallel corpora. In: Proc. of the 1st WDE (2009)
8. Gliozzo, A.M., Ranieri, M.: Crossing parallel corpora and multilingual lexical databases for wsd. In: Proc. of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (2005)
9. Makki, R., Homayounpour, M.: Word sense disambiguation of farsi homographs using thesaurus and corpus. In: Proc. of the 6th International Conference on Advances in Natural Language Processing (2008)
10. Soltani, M., Faili, H.: A statistical approach on persian word sense disambiguation. In: The 7th International Conference on INFOS (2010)
11. Shamsfard, M.: Semi automatic development of farsnet; the persian wordnet. In: Proc. of 5th Global WordNet Conference (2010)
12. Miangah, T.: Constructing a large-scale english-persian parallel corpus. Meta: Translators' Journal (2009)
13. Pilevar, T., Faili, H.: Persiansmt: A first attempt to english-persian statistical machine translation. In: JADT (2010)
14. Shamsfard, M.: Step-1: Standard text preparation for persian language. In: Proc. of Machine Translation Summit XII (2009)
15. Pedersen, T., Kolhatkar, V.: Wordnet:senserelate:allwords: a broad coverage word sense tagger that maximizes semantic relatedness. In: Proc. of Human Language Technologies: NAACL, Companion Volume: Demonstration Session (2009)
16. Banerjee, S.: Extended gloss overlaps as a measure of semantic relatedness. In: Proc. of the 18th International Joint Conference on Artificial Intelligence (2003)
17. Specia, L., et.al.: An automatic approach to create a sense tagged corpus for word sense disambiguation in machine translation. In: Proc. of the 2nd Meaning Workshop (2005)
18. Resnik, P., Yarowsky, D.: Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. Nat. Lang. Eng. (1999)

Extracting Product Descriptions from Polish E-Commerce Websites Using Classification and Clustering*

Piotr Kołaczkowski and Piotr Gawrysiak

Institute of Computer Science, Warsaw University of Technology

Abstract. A novel method for extracting product descriptions from e-commerce websites is presented. The algorithm consists of three major steps: (1) extracting descriptions of appropriate length from the source documents related to the search query using shallow text analysis methods; (2) assigning each of the description to one of the predefined categories by means of text classification and (3) grouping the results by a text clustering algorithm to return the descriptions found in the clusters with the highest quality. The recall and precision of the search are examined using a set of queries for laptops currently being sold in popular shopping sites. It is shown that, although the extraction method based purely on the classification and the method based purely on the clustering give acceptable results, the highest precision is achieved when using them together. It was also observed that examining about 20 first sites returned by Google is sufficient to get high quality descriptions of popular products.

1 Introduction

Recent years brought us an explosive growth of Internet usage in commerce, both in the form of business to consumer and business to business applications. While such quick pace of development of services has been highly beneficial to customers and economy, the technical quality of the communications infrastructure, that powers contemporary electronic or Internet stores, has not evolved with the same speed. Instead of systems, exchanging highly structured information, as envisioned by Semantic Web enthusiasts, we still have to deal with traditional natural language information resources, providing financial data and product information in fuzzy form. While this usually does not create too many problems for an average customer just browsing the web in pursuit of next gadget, it makes any kind of aggregate information processing difficult or downright impossible. A typical price comparison engine such as Google Shopping or Ceneo constitutes a good example. Lack of structured protocols and data formats used

* This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 456–464, 2011.

for storing product information results in multitude of factual errors in their descriptions on ecommerce sites and problems with such a seemingly simple task, as identification of similar products.

Historically there were two approaches to taming the problem of unstructured data. One of them is getting rid of said data in favor of information in more rigid form – an approach that clearly has not been possible in the case of commercial Web, as mentioned above. Another one involves using artificial intelligence methods, such as knowledge discovery algorithms, in order to automatically – at least partially – extract and arrange useful data from unstructured source. Such approach was a basis of a research project that our team undertook for a business client, looking for a way of effectively extracting and comparing product descriptions and product technical specifications form various online resources. The preliminary results of this project are described in this paper.

The paper is structured as follows: second chapter provides information about some previous attempts at information extraction and structuring from web resources. Next section presents an outline of data extraction and mining algorithms that we devised. In chapter 4 experimental results of its evaluation are presented and discussed, while the final chapter contains closing remarks.

2 Related Work

Many algorithms described in the literature concentrate on automatic or semi-automatic creation of rules for extracting structured information from the webpages and then applying these rules to extract the desired information. The rules can be created manually, by creating a special wrapper program or automatically, by machine learning techniques. Manual wrapper creation requires the user to first study a number of webpages and analyse their layout and then to to create a wrapper program, performing the actual extraction. Although there exist a number of toolkits that provide some support in form of pattern specification languages or automatic code generation [7,11,14], the process of creating the wrapper manually is time consuming and error-prone. Obviously, this approach does not scale to a large number of different webpages. Additionally, if the layout of the pages is changed, the wrappers must be recreated from scratch.

Therefore, much research has been done in the area of semi-automatic and automatic wrapper generation. The first approach requires preparing a small set of webpages and labelling the appropriate interesting fragments in them. Thanks to labelling, by analysing the structure of the webpage around the labelled fragments, the computer program can observe common patterns and automatically discover the rules to extract the desired information. The rules can be then applied to massively extract information from other pages, if only they have similar layout to the ones in the training set. Examples of such systems are described in [4,6,9,10,12,13]. Instead of learning rules from the training set, algorithms [2,15] use instance-based methods, similar to the k-nearest-neighbors approach. Whenever information needs to be extracted from a new page, the page is compared to the pages previously successfully processed. Thus, only a single manually labelled page is required to start the process of data extraction. Only if the new

page cannot be matched with any previously labelled instance, manual labelling is required.

The best scalability is offered by the methods requiring no manual labelling of pages [3,5]. The algorithm [3] detect patterns in pages containing repeated items, e.g. search results or product lists. Because of repetitions, the HTML code forming the page template can be separated from the text being extracted. The method [5] achieves the same by analysing similarities and differences between a set of pages sharing the same template. The algorithm presented in this paper also belongs to this class of algorithms. However, our algorithm concentrates more on what should be extracted by means of classification, instead of discovering common template fragments. It also does not assume existence of any regularity or patterns in pages. Therefore it can be applied to random results returned by the search engine.

3 Algorithm

As the input, the presented system takes a query in form of a few terms identifying the product. Typically, this should be the vendor name and the model name. For the best results, the query should identify exactly one product, and not a whole family of products varying in features. As the output, the system should return descriptions that characterize the product it has been asked for. The descriptions have to be found in the WWW. An example description is presented in Figure 1.

The query is first used to find the related webpages. This can be done by any of the well known full-text-search methods. In our implementation, for simplicity of implementation, we used the Google Search API. Google returns a list of

Fig. 1. Example e-commerce page displaying a description of a product

Internet addresses of the documents, which then have to be fully downloaded. Initially, we took not more than the first 50 search results for further processing, but taking less proved to be sufficient.

The index may return the same document stored under different URLs. The documents are clustered using agglomerative hierarchical clustering algorithm [8] and cosine similarity to eliminate the duplicates. Two documents a and b with $cos\theta(a, b) > 0.95$ are regarded as the same document. Duplicate documents need to be removed in order not to mislead the cluster analysis performed later, which assumes that product descriptions come from different sources.

Each of the remaining documents undergoes shallow text analysis, to filter out the undesired content like HTML tags, scripts, images and split the documents into smaller *fragments* that might form the product descriptions. Most of the sites are usually designed to look good and to provide good user experience. The semantic structure of the documents is the secondary concern. Website designers use various techniques for formatting the content, not always following the recommendations of World Wide Web Consortium (W3C). For example, the page can be laid out as a large, invisible table, or the content visible to the user as a table can be created with a separate section (div) for each of the rows. Therefore, simply dividing the content by paragraphs or the lowest level sections, may produce fragments containing only parts of the product description. Instead, we build the DOM representation of the document, and, starting from the lowest level, recursively merge the nodes, until fragments of desired minimum length are formed. This process is illustrated in Figure 2. If a node is selected, the merging of this branch stops, and the ascendants of the selected node are not considered. Thus, a single node cannot be used to build more than one fragment, and there is no risk of introducing duplicates.

To improve the quality of the classification and clustering algorithms used further, the stop-words are filtered out from the fragments, based on a fixed list of 274 Polish stop-words. Because Polish is a strongly inflected language, the remaining words are replaced by their lemmas. A dictionary-based approach is

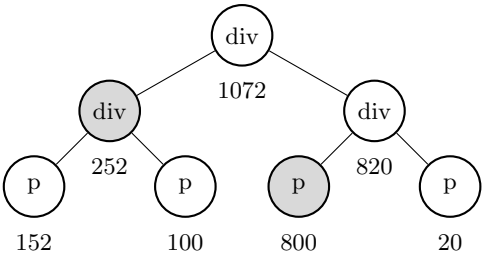


Fig. 2. DOM element merging technique used for splitting the page content into fragments. The nodes represent DOM tree elements and the labels below them represent the length of their textual representation. The lowest elements containing text longer than 200 characters are selected.

used. The dictionary has been made by the players of the Polish word game called "Literaki" [1] (very similar to Scrabble), and it contains about 3.5 million various word forms.

The fragments are then classified. There is one class per each named *product category*. The list of product categories is configured by the administrator of the system, and it reflects the list of product categories found in the e-commerce sites that are explored. There is also a special class called *other*, which groups the fragments that could not be classified as belonging to any other product class. Fragments not describing any product are expected to fall into that class. Any known classification algorithm can be used. The selection of a classifier affects the quality of obtained results, the performance of the system and the amount of work required to prepare the system before it can be used. Some classifiers require learning process, like K Nearest Neighbours, Artificial Neural Networks or Support Vector Machine classifiers [8]. Although they provide high quality results, we decided to use a simpler approach, that does not require careful training set selection phase and allows for describing product categories by lists of weighted keywords. The weight of a keyword expresses the likelyhood that the keyword is used to describe a product belonging to the category. Weights can be negative. A negative weight of the keyword means that the fragment containing this keyword is unlikely to belong to the category. Given a fragment f, a set of keywords K with their weights $w : K \to R$, total number of words $N(f)$ in the fragment f, and number of occurences $n(k, f)$ of word k in the fragment f, we can formulate the likelyhood p that the fragment f belongs to the category described by keywords K as:

$$p(f, K) = \frac{\sum_{k \in K} \operatorname{sgn} w(k) |w(k) n(k, f)|^{\alpha}}{N(f)} \tag{1}$$

The $\alpha > 0$ parameter controls how sensitive is the p function to the number of repetitions of the same keyword. For $\alpha = 1$, the result is proportional to the number of occurences of the keyword. Therefore, there is no difference, whether the fragment contains the same keyword twice, or two different keywords (assuming same weights). However, for smaller α, a fragment containing different keywords is valued higher. We empirically checked that when using α values less than 0.5 we were able to create classifiers with acceptable quality. Each fragment is assigned to the category for which the formula (1) yields the highest value. This highest value of p is further referred to as a *score* $p_{\mathrm{sel}}(f)$ of the fragment. Fragments with p value lower than some threshold value $p_{\min}$ are classified as belonging to the *other* class and discarded from further processing. If there are too many fragments at this point, to save processing time, m fragments with the highest score are chosen.

The remaining fragments are grouped into clusters by an agglomerative hierarchical clustering algorithm. Each fragment is represented as a term vector, where the terms are weighted using tf-idf. As a distance function between two term vectors, cosine distance $\Delta(f_1, f_2) = 1 - \cos\theta(f_1, f_2)$ has been chosen. Clusters are merged basing on single-linkage. Fragments classified as belonging to different categories are not merged. Merging stops when the distance between

the two nearest clusters is higher than some fixed threshold value $d_{\max}$. Each cluster C_j is assigned a score calculated by the following formula:

$$s_j = \frac{|C_j|^2(|C_j|-1)}{\sum_{a,b\in C_j} \Delta^2(a,b)} \sum_{f\in C_j} p_{\text{sel}}(f) \tag{2}$$

The score increases with the size of the cluster. Obtaining a large cluster means that there are many different pages containing similar information on the searched product, therefore it can be believed that such information has higher credibility. Also, the more similar are the fragments in the cluster, the higher is the likelyhood that they describe the same product, thus the higher is the score. The cluster score is also proportional to the sum of the individual scores of the contained fragments, calculated by the classifier. The user is presented the medoids of the clusters with the scores exceeding a constant minimum threshold. The higher is this threshold set, the better is the precision and the lower is the recall of the algorithm.

4 Experimental Results

We evaluated precision and recall of the proposed algorithm in four variants: (1) the classification and grouping used together (CG); (2) fragments after the classification directly used as the output, without further grouping (C); (3) without the classification stage and with all fragment scores set to 1.0 (G) and (4) no classification and no clustering at all, with the algorithm reporting result fragments immediately after the DOM merging phase (M). In each of the experiments, the initial document set retrieved from Google was limited to 50 documents, m was set to 120 fragments and the output result set size was limited to 10 fragments. These settings allowed to answer most queries within less than 30 seconds when using the CG version of the algorithm, including the time required for asking Google and download the pages.

The returned fragments were then compared by an expert to the original specification of the product. The results are presented in Table 1. The classification and clustering applied together provided the highest search precision. Over 80% of returned fragments contained descriptions of the searched product. Turning off the clustering increased the total number of returned fragments, but decreased the overall precision. Surprisingly, skipping the classification phase not only decreases the precision, but also reduces the total number of returned fragments. This is caused by limiting the number of fragments entering the clustering phase. Without the classification phase, the average quality of these fragments is poor. Only about quarter of these fragments were valid product descriptions in the experiment, as can be seen in the M column of Table 1. Therefore, the probability to form highly-scored clusters from these fragments is lower, than from the fragments filtered in the classification phase.

Apart from performing the quantitive precision analysis, we looked closer at the content of the returned false positive fragments. They can be divided into the following two categories:

- Fragments that are formally descriptions of some products, but describe not the products that the user searched for. This kind of false positive was mostly reported when using classification without clustering. A single e-commerce page may reference more than one product. For example, a product details page may additionally contain some other product recommendations or advertisements. Because the returned fragments need not contain the query terms, wrong fragment of such page can be returned. False positives of this kind are usually filtered out in the clustering phase, because the probablity of finding more fragments describing exactly the same irrelevant product is low, and such fragments cannot form high quality clusters.
- Navigation menus and other common elements of the website template. Because these elements are common across many pages of the same website they can form large clusters and be highly scored by the clustering algorithm. Most of false positives of this kind are filtered out in the classification phase.

Precision is influenced by the number of pages given at the input. If the number of pages actually containing relevant product desctriptions is low, both precision and recall decrease and results become unuseful. We have found that 10 pages is a minimum required to get at least one useful product description.

5 Final Notes

The algorithm presented in this paper has shown itself useful for searching for descriptions of products in e-commerce sites. Within a few seconds, it can answer queries with good precision, significantly reducing the amount of work required for manually finding product descriptions in the web. It requires only a constant amount of preparatory work for training or configuring the classifier, therefore it does not suffer from scalability issues of algorithms based on page labelling. Additionally it does not rely on template-generated pages nor any special patterns

Table 1. Number of returned positives (P) and true positives (TP) in four variants of the algorithm (details in the text)

Query	CG		C		G		M	
	P	TP	P	TP	P	TP	P	TP
Dell Vostro 3500	3	3	9	8	5	3	10	3
Dell Vostro 3700	5	5	6	6	5	3	10	2
Dell Latitude E6400	4	2	8	3	1	1	10	2
Lenovo ThinkPad R500	4	4	9	6	5	1	10	1
IBM Lenovo G560	6	6	8	6	4	2	6	1
Acer Aspire 5741	6	4	10	5	3	2	10	4
Toshiba Satellite A660	7	5	7	3	1	1	10	4
Mean precision:	0,83		0,65		0,54		0,26	

in the page content. These features make it great for supporting e-commerce site content editors by allowing them to quickly locate good descriptions of products they enter into the system.

We presume that using better, more complex classification algorithms could lead to increase of precision and recall. Also merging the presented technique with methods employing pattern discovery might offer further quality enhancements.

References

1. Literaki online, http://www.kurnik.pl/literaki/
2. Chang, C.H., Kuo, S.C.: OLERA: Semisupervised web-data extraction with visual support. IEEE Intelligent Systems 19, 56–64 (2004), http://dx.doi.org/10.1109/MIS.2004.71
3. Chang, C.H., Lui, S.C.: IEPAD: information extraction based on pattern discovery. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 681–688. ACM, New York (2001), http://doi.acm.org/10.1145/371920.372182
4. Cohen, W.W., Hurst, M., Jensen, L.S.: A flexible learning system for wrapping tables and lists in html documents. In: Proceedings of the 11th International Conference on World Wide Web, WWW 2002, pp. 232–241. ACM, New York (2002), http://doi.acm.org/10.1145/511446.511477
5. Crescenzi, V., Mecca, G., Merialdo, P.: RoadRunner: Towards automatic data extraction from large web sites. In: Proceedings of the 27th International Conference on Very Large Data Bases, VLDB 2001, pp. 109–118. Morgan Kaufmann Publishers Inc., San Francisco (2001)
6. Freitag, D., Kushmerick, N.: Boosted wrapper induction. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 577–583. AAAI Press, Menlo Park (2000), http://portal.acm.org/citation.cfm?id=647288.723413
7. Hammer, J., Garcia-molina, H., Cho, J., Aranha, R., Crespo, A.: Extracting semistructured information from the web. In: In Proceedings of the Workshop on Management of Semistructured Data, pp. 18–25 (1997)
8. Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco (2005)
9. Knoblock, C.A., Lerman, K., Minton, S., Muslea, I.: Accurately and reliably extracting data from the web: a machine learning approach. In: Szczepaniak, P.S., Segovia, J., Kacprzyk, J., Zadeh, L.A. (eds.) Intelligent Exploration of the Web, pp. 275–287. Physica-Verlag GmbH, Heidelberg (2003), http://portal.acm.org/citation.cfm?id=941713.941732
10. Kushmerick, N.: Wrapper induction for information extraction. Ph.D. thesis, University of Washington (1997)
11. Liu, L., Pu, C., Han, W.: XWRAP: An XML-enabled wrapper construction system for web information sources. In: Proceedings of the 16th International Conference on Data Engineering, pp. 611–621. IEEE Computer Society, Washington, DC, USA (2000), http://portal.acm.org/citation.cfm?id=846219.847340
12. Muslea, I., Minton, S., Knoblock, C.: A hierarchical approach to wrapper induction. In: Proceedings of the Third Annual Conference on Autonomous Agents,AGENTS 1999, pp. 190–197. ACM, New York (1999), http://doi.acm.org/10.1145/301136.301191

13. Pinto, D., McCallum, A., Wei, X., Croft, W.B.: Table extraction using conditional random fields. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 235–242. ACM, New York (2003), http://doi.acm.org/10.1145/860435.860479
14. Sahuguet, A., Azavant, F.: WYSIWYG web wrapper factory (W4F). In: Proceedings of WWW Conference (1999)
15. Zhai, Y., Liu, B.: Extracting web data using instance-based learning. World Wide Web 10, 113–132 (2007), http://portal.acm.org/citation.cfm?id=1265159.1265174

Cut-Free ExpTime Tableaux for Checking Satisfiability of a Knowledge Base in the Description Logic $\mathcal{ALCI}$

Linh Anh Nguyen

Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
nguyen@mimuw.edu.pl

Abstract. We give the first direct cut-free ExpTime (optimal) tableau decision procedure, which is not based on transformation or on the pre-completion technique, for checking satisfiability of a knowledge base in the description logic $\mathcal{ALCI}$.

1 Introduction

Description logics (DLs) are used, amongst others, as a logical base for the Web Ontology Language OWL. They represent the domain of interest in terms of concepts, individuals, and roles. A concept is interpreted as a set of individuals, while a role is interpreted as a binary relation among individuals. A knowledge base in a DL usually has two parts: a TBox consisting of terminology axioms, and an ABox consisting of assertions about individuals. One of the basic inference problems in DLs, which we denote by Sat, is to check satisfiability of a knowledge base. Other inference problems in DLs are usually reducible to this problem. For example, the problem of checking consistency of a concept w.r.t. a TBox (further denoted by $Cons$) is linearly reducible to Sat.

In this paper we study automated reasoning in the description logic $\mathcal{ALCI}$, which extends the basic description logic $\mathcal{ALC}$ with inverse roles. Both problems, Sat and $Cons$, in $\mathcal{ALCI}$ are ExpTime-complete. To deal with these problems one can translate them into another problem in another logic, for example, by encoding the ABox by "nominals" and "internalizing" the TBox or by using other transformations [2,1]. Then one can use an available decision procedure for the latter problem. This approach is, however, not efficient in practice.[1] Direct decision procedures for DLs are usually based on tableaux and have been highly optimized. Traditional tableau decision procedures for DLs use backtracking to deal with "or"-branchings and are sub-optimal in terms of worst-case complexity (e.g. 2ExpTime instead of ExpTime). Together with Goré and Szałas we have developed direct complexity-optimal tableau decision procedures for a number

[1] In the well-known tutorial [6], Horrocks and Sattler wrote *"direct algorithm / implementation instead of encodings"* and *"even simple domain encoding is disastrous with large numbers of roles"*.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 465–475, 2011.

of modal and description logics by using global caching [3,9,12,10,11,13]. For the logics $\mathcal{SHI}$ [3], CPDL [10] and REG^c [11] we used analytic cut rules to deal with inverse roles and converse modal operators. As cuts are not efficient in practice, Goré and Widmann developed cut-free ExpTime tableau decision procedures, based on global state caching, for the *Cons* problem in $\mathcal{ALCI}$ [4] and CPDL [5]. They did not study the more general problem *Sat* for these logics.

In this paper we give the first direct cut-free ExpTime (optimal) tableau decision procedure, which is not based on transformation or on the pre-completion technique, for the *Sat* problem in $\mathcal{ALCI}$. We use our methods of [9,12,13] to deal with ABoxes and a similar idea as of [4,5] to deal with inverse roles. Our procedure can be implemented with various optimizations as in [7].

2 Notation and Semantics of $\mathcal{ALCI}$

Our language uses a finite set $\mathbf{C}$ of *concept names*, a finite set $\mathbf{R}$ of role names, and a finite set $\mathbf{I}$ of individual names. We use letters like A and B for *concept names*, r and s for *role names*, and a and b for *individual names*. We refer to A and B also as *atomic concepts*, and to a and b as *individuals*.

For $r \in \mathbf{R}$, let r^- be a new symbol, called the *inverse* of r. Let $\mathbf{R}^- = \{r^- \mid r \in \mathbf{R}\}$ be the set of *inverse roles*. A *role* is any member of $\mathbf{R} \cup \mathbf{R}^-$. We use letters like R and S for roles. For $r \in \mathbf{R}$, define $(r^-)^- = r$.

Concepts in $\mathcal{ALCI}$ are formed using the following BNF grammar:

$$C, D ::= \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

We use letters like C and D to denote arbitrary concepts.

A *TBox* is a finite set of axioms of the form $C \sqsubseteq D$ or $C \doteq D$. An *ABox* is a finite set of *assertions* of the form $a\!:\!C$ (*concept assertion*) or $R(a, b)$ (*role assertion*). A *knowledge base* in $\mathcal{ALCI}$ is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox.

A *formula* is defined to be either a concept or an ABox assertion. We use letters like φ, ψ, ξ to denote formulas, and letters like X, Y, Γ to denote sets of formulas.

An *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ consists of a non-empty set $\Delta^{\mathcal{I}}$, called the *domain* of $\mathcal{I}$, and a function $\cdot^{\mathcal{I}}$, called the *interpretation function* of $\mathcal{I}$, that maps every concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, maps every role name r to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and maps every individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation function is extended to inverse roles and complex concepts as follows:

$$(r^-)^{\mathcal{I}} = \{\langle x, y\rangle \mid \langle y, x\rangle \in r^{\mathcal{I}}\} \qquad \top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad \bot^{\mathcal{I}} = \emptyset$$

$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

$$(\forall R.C)^{\mathcal{I}} = \big\{x \in \Delta^{\mathcal{I}} \mid \forall y\big[(x, y) \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\big]\big\}$$

$$(\exists R.C)^{\mathcal{I}} = \big\{x \in \Delta^{\mathcal{I}} \mid \exists y\big[(x, y) \in R^{\mathcal{I}} \text{ and } \ y \in C^{\mathcal{I}}\big]\big\}$$

Note that $(r^-)^{\mathcal{I}} = (r^{\mathcal{I}})^{-1}$ and this is compatible with $(r^-)^- = r$.

For a set Γ of concepts, define $\Gamma^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}} \text{ for all } C \in \Gamma\}$.

An interpretation $\mathcal{I}$ is a *model of a TBox* $\mathcal{T}$ if for every axiom $C \sqsubseteq D$ (resp. $C \doteq D$) of $\mathcal{T}$, we have that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (resp. $C^{\mathcal{I}} = D^{\mathcal{I}}$).

An interpretation $\mathcal{I}$ is a *model of an ABox* $\mathcal{A}$ if for every assertion $a\!:\!C$ (resp. $R(a,b)$) of $\mathcal{A}$, we have that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$).

An interpretation $\mathcal{I}$ is a *model of a knowledge base* $(\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ is a model of both $\mathcal{T}$ and $\mathcal{A}$. A knowledge base $(\mathcal{T}, \mathcal{A})$ is *satisfiable* if it has a model.

An interpretation $\mathcal{I}$ *satisfies* a concept C (resp. a set X of concepts) if $C^{\mathcal{I}} \neq \emptyset$ (resp. $X^{\mathcal{I}} \neq \emptyset$). A set X of concepts is *satisfiable w.r.t. a TBox* $\mathcal{T}$ if there exists a model of $\mathcal{T}$ that satisfies X. For $X = Y \cup \mathcal{A}$, where Y is a set of concepts and $\mathcal{A}$ is an ABox, we say that X is *satisfiable w.r.t. a TBox* $\mathcal{T}$ if there exists a model of $\mathcal{T}$ and $\mathcal{A}$ that satisfies X.

3 A Tableau Calculus for $\mathcal{ALCI}$

We assume that concepts and ABox assertions are represented in negation normal form (NNF), where $\neg$ occurs only directly before atomic concepts.[2] We use $\overline{C}$ to denote the NNF of $\neg C$, and for $\varphi = a\!:\!C$, we use $\overline{\varphi}$ to denote $a\!:\!\overline{C}$. For simplicity, we treat axioms of $\mathcal{T}$ as concepts representing global assumptions: an axiom $C \sqsubseteq D$ is treated as $\overline{C} \sqcup D$, while an axiom $C \doteq D$ is treated as $(\overline{C} \sqcup D) \sqcap (\overline{D} \sqcup C)$. That is, we assume that $\mathcal{T}$ consists of concepts in NNF. Thus, an interpretation $\mathcal{I}$ is a model of $\mathcal{T}$ iff $\mathcal{I}$ validates every concept $C \in \mathcal{T}$. As this way of handling the TBox is not efficient in practice, the absorption technique like the one discussed in [9,13] can be used to improve the performance of our algorithm.

From now on, let $(\mathcal{T}, \mathcal{A})$ be a knowledge base in NNF of the logic $\mathcal{ALCI}$. In this section we present a tableau calculus for checking satisfiability of $(\mathcal{T}, \mathcal{A})$.

In what follows we define tableaux as rooted "and-or" graphs. Such a graph is a tuple $G = (V, E, \nu)$, where V is a set of nodes, $E \subseteq V \times V$ is a set of edges, $\nu \in V$ is the root, and each node $v \in V$ has a number of attributes. If there is an edge $(v, w) \in E$ then we call v a *predecessor* of w, and call w a *successor* of v. The set of all attributes of v is called the *contents of* v. Attributes of tableau nodes are:

- $\mathit{Type}(v) \in \{\mathsf{state}, \mathsf{non\text{-}state}\}$. If $\mathit{Type}(v) = \mathsf{state}$ then we call v a *state*, else we call v a *non-state* (or an *internal* node). If $\mathit{Type}(v) = \mathsf{state}$ and $(v, w) \in E$ then $\mathit{Type}(w) = \mathsf{non\text{-}state}$.
- $\mathit{SType}(v) \in \{\mathsf{complex}, \mathsf{simple}\}$ is called the subtype of v. If $\mathit{SType}(v) = \mathsf{complex}$ then we call v a *complex node*, else we call v a *simple node*. The graph never contains edges from a simple node to a complex node. If (v, w) is an edge from a complex node v to a simple node w then $\mathit{Type}(v) = \mathsf{state}$ and $\mathit{Type}(w) = \mathsf{non\text{-}state}$. The root of the graph is a complex node.
- $\mathit{Status}(v) \in \{\mathsf{unexpanded}, \mathsf{expanded}, \mathsf{incomplete}, \mathsf{unsat}, \mathsf{sat}\}$.

[2] Every formula can be transformed to an equivalent formula in NNF.

- $\mathit{Label}(v)$ is a finite set of formulas, called the label of v. The label of a complex node consists of ABox assertions, while the label of a simple node consists of concepts.
- $\mathit{RFmls}(v)$ is a finite set of formulas, called the set of reduced formulas of v.
- $\mathit{DFmls}(v)$ is a finite set of formulas, called the set of disallowed formulas of v.
- $\mathit{StatePred}(v) \in V \cup \{\mathsf{null}\}$ is called the state-predecessor of v. It is available only when $\mathit{Type}(v) = \mathsf{non\text{-}state}$. If v is a non-state and G has no paths connecting a state to v then $\mathit{StatePred}(v) = \mathsf{null}$. Otherwise, G has exactly one state u that is connected to v via a path not containing any other states. In that case, $\mathit{StatePred}(v) = u$.
- $\mathit{ATPred}(v) \in V$ is called the after-transition-predecessor of v. It is available only when $\mathit{Type}(v) = \mathsf{non\text{-}state}$. If v is a non-state and $v_0 = \mathit{StatePred}(v)$ ($\neq \mathsf{null}$) then there is exactly one successor v_1 of v_0 such that every path connecting v_0 to v must go through v_1, and we have that $\mathit{ATPred}(v) = v_1$. We define $\mathtt{AfterTrans}(v) = (\mathit{ATPred}(v) = v)$. If $\mathtt{AfterTrans}(v)$ holds then either v has no predecessors (i.e. it is the root of the graph) or it has exactly one predecessor u and u is a state.
- $\mathit{CELabel}(v)$ is a formula called the coming edge label of v. It is available only when v is a successor of a state u (and $\mathit{Type}(v) = \mathsf{non\text{-}state}$). In that case, we have $u = \mathit{StatePred}(v)$, $\mathtt{AfterTrans}(v)$ holds, $\mathit{CELabel}(v) \in \mathit{Label}(u)$, and
  - if $\mathit{SType}(u) = \mathsf{simple}$ then $\mathit{CELabel}(v)$ is of the form $\exists R.C$ and $C \in \mathit{Label}(v)$
  - else $\mathit{CELabel}(v)$ is of the form $a{:}\exists R.C$ and $C \in \mathit{Label}(v)$.

  Informally, v was created from u to realize the formula $\mathit{CELabel}(v)$ at u.
- $\mathit{ConvMethod}(v) \in \{0, 1\}$ is called the converse method of v. It is available only when $\mathit{Type}(v) = \mathsf{state}$.
- $\mathit{FmlsRC}(v)$ is a set of formulas, called the set of formulas required by converse for v. It is available only when $\mathit{Type}(v) = \mathsf{state}$ and will be used only when $\mathit{ConvMethod}(v) = 0$.
- $\mathit{AltFmlSetsSC}(v)$ is a set of sets of formulas, called the set of alternative sets of formulas suggested by converse for v. It is available only when $\mathit{Type}(v) = \mathsf{state}$ and will be used only when $\mathit{ConvMethod}(v) = 1$.
- $\mathit{AltFmlSetsSCP}(v)$ is a set of sets of formulas, called the set of alternative sets of formulas suggested by converse for the predecessor of v. It is available only when v has a predecessor being a state and will be used only when $\mathit{ConvMethod}(v) = 1$.

We define

$$\mathtt{AFmls}(v) = \mathit{Label}(v) \cup \mathit{RFmls}(v)$$

$$\mathtt{Kind}(v) = \begin{cases} \mathsf{and\text{-}node} \text{ if } \mathit{Type}(v) = \mathsf{state} \\ \mathsf{or\text{-}node} \text{ if } \mathit{Type}(v) = \mathsf{non\text{-}state} \end{cases}$$

$$\mathtt{BeforeFormingState}(v) = v \text{ has a successor which is a state}$$

The set $\mathtt{AFmls}(v)$ is called the available formulas of v. In an "and-or" graph, states play the role of *"and"-nodes*, while non-states play the role of *"or"-nodes*.

$(\sqcap)\ \dfrac{X, C \sqcap D}{X, C, D}$	$(\sqcup)\ \dfrac{X, C \sqcup D}{X, C \mid X, D}$
$(\exists)\ \dfrac{X, \exists R_1.C_1, \ldots, \exists R_k.C_k}{C_1, X_1, \mathcal{T}\ \&\ \ldots\ \&\ C_k, X_k, \mathcal{T}}$ if $\begin{cases} X \text{ contains no concepts of the} \\ \text{form } \exists R.D \text{ and, for } 1 \leq i \leq k, \\ X_i = \{D \mid \forall R_i.D \in X\} \end{cases}$	
$(\sqcap')\ \dfrac{X,\ a{:}(C \sqcap D)}{X,\ a{:}C,\ a{:}D}$	$(\sqcup')\ \dfrac{X,\ a{:}(C \sqcup D)}{X,\ a{:}C \mid X,\ a{:}D}$
$(\forall')\ \dfrac{X,\ a{:}\forall R.C,\ R(a,b)}{X,\ a{:}\forall R.C,\ R(a,b),\ b{:}C}$	$(\forall_i')\ \dfrac{X,\ a{:}\forall R.C,\ R^-(b,a)}{X,\ a{:}\forall R.C,\ R^-(b,a),\ b{:}C}$
$(\exists')\ \dfrac{X,\ a_1{:}\exists R_1.C_1, \ldots, a_k{:}\exists R_k.C_k}{C_1, X_1, \mathcal{T}\ \&\ \ldots\ \&\ C_k, X_k, \mathcal{T}}$ if $\begin{cases} X \text{ contains no assertions of the} \\ \text{form } a{:}\exists R.D \text{ and, for } 1 \leq i \leq k, \\ X_i = \{D \mid a_i{:}\forall R_i.D \in X\} \end{cases}$	

Table 1. Some rules of the tableau calculus $C_{\mathcal{ALCI}}$

By the *local graph* of a state v we mean the subgraph of G consisting of all the path starting from v and not containing any other states. Similarly, by the local graph of a non-state v we mean the subgraph of G consisting of all the path starting from v and not containing any states.

We apply global state caching: if v_1 and v_2 are different states then $\mathit{Label}(v_1) \neq \mathit{Label}(v_2)$ or $\mathit{RFmls}(v_1) \neq \mathit{RFmls}(v_2)$ or $\mathit{DFmls}(v_1) \neq \mathit{DFmls}(v_2)$. If v is a non-state such that $\texttt{AfterTrans}(v)$ holds then we also apply global caching for the local graph of v: if w_1 and w_2 are different nodes of the local graph of v then $\mathit{Label}(w_1) \neq \mathit{Label}(w_2)$ or $\mathit{RFmls}(w_1) \neq \mathit{RFmls}(w_2)$ or $\mathit{DFmls}(w_1) \neq \mathit{DFmls}(w_2)$.

Our calculus $C_{\mathcal{ALCI}}$ for the description logic $\mathcal{ALCI}$ will be specified, amongst others, by a finite set of tableau rules, which are used to expand nodes of tableaux. A *tableau rule* is specified with the following information: the kind of the rule (an "and"-rule or an "or"-rule); the conditions for applicability of the rule (if any); the priority of the rule; the number of successors of a node resulting from applying the rule to it, and the way to compute their contents.

Tableau rules are usually written downwards, with a set of formulas above the line as the *premise*, which represents the label of the node to which the rule is applied, and a number of sets of formulas below the line as the *(possible) conclusions*, which represent the labels of the successor nodes resulting from the application of the rule. Possible conclusions of an "or"-rule are separated

Function NewSucc$(v, type, sType, ceLabel, label, rFmls, dFmls)$

Global data: a rooted graph (V, E, ν).
Purpose: create a new successor for v.

create a new node w, $V := V \cup \{w\}$, **if** $v \neq$ null **then** $E := E \cup \{(v, w)\}$;
$Type(w) := type$, $SType(w) := sType$, $Status(w) :=$ unexpanded;
$Label(w) := label$, $RFmls(w) := rFmls$, $DFmls(w) := dFmls$;
if $type =$ non-state **then**
  if $v =$ null *or* $Type(v) =$ state **then** $StatePred(w) := v$, $ATPred(w) := w$
  else $StatePred(w) := StatePred(v)$, $ATPred(w) := ATPred(v)$;
  if $Type(v) =$ state **then** $CELabel(w) := ceLabel$, $AltFmlSetsSCP(w) := \emptyset$
else $ConvMethod(w) := 0$, $FmlsRC(w) := \emptyset$, $AltFmlSetsSC(w) := \emptyset$;
return w

Function FindProxy$(type, sType, v_1, label, rFmls, dFmls)$

Global data: a rooted graph (V, E, ν).

if $type =$ state **then** $W := V$ **else** $W :=$ the nodes of the local graph of v_1;
if *there exists* $w \in W$ *such that* $Type(w) = type$ *and* $SType(w) = sType$ *and* $Label(w) = label$ *and* $RFmls(w) = rFmls$ *and* $DFmls(w) = dFmls$ **then return** w
else return null

Function ConToSucc$(v, type, sType, ceLabel, label, rFmls, dFmls)$

Global data: a rooted graph (V, E, ν).
Purpose: connect v to a successor, which is created if necessary.

if $type =$ state **then** $v_1 :=$ null **else** $v_1 := ATPred(v)$
$w :=$ FindProxy$(type, sType, v_1, label, rFmls, dFmls)$;
if $w \neq$ null **then** $E := E \cup \{(v, w)\}$
else $w :=$ NewSucc$(v, type, sType, ceLabel, label, rFmls, dFmls)$;
return w

Function TUnsat(v)

return *(*$\bot \in Label(v)$ *or there exists* $\{\varphi, \overline{\varphi}\} \subseteq Label(v)$*)*

Function TSat(v)

return *(*$Status(v) =$ unexpanded *and no rule except* $(conv)$ *is applicable to* v*)*

Function ToExpand

Global data: a rooted graph (V, E, ν).

if *there exists a node* $v \in V$ *with* $Status(v) =$ unexpanded **then return** v
else return null

by |, while conclusions of an "and"-rule are separated by &. If a rule is a unary rule (i.e. a rule with only one possible conclusion) or an "and"-rule then its conclusions are "firm" and we ignore the word "possible". The meaning of an "or"-rule is that if the premise is satisfiable w.r.t. $\mathcal{T}$ then some of the possible conclusions are also satisfiable w.r.t. $\mathcal{T}$, while the meaning of an "and"-rule is

Procedure `Apply`(ρ, v)

Global data: a rooted graph (V, E, ν).
Input: a rule ρ and a node $v \in V$ s.t. if $\rho \neq (conv)$ then $Status(v) = \mathsf{unexpanded}$ else $Status(v) = \mathsf{expanded}$ and `BeforeFormingState`(v) holds.
Purpose: applying the tableau rule ρ to the node v.

```
1  if ρ = (forming-state) then
2  |  ConToSucc(v, state, SType(v), null, Label(v), RFmls(v), DFmls(v))
3  else if ρ = (conv) then ApplyConvRule(v) // defined on page 472
4  else if ρ ∈ {(∃), (∃′)} then
5  |  ApplyTransRule(ρ, v); // defined on page 472
6  |  if Status(v) = {incomplete, unsat, sat} then
7  |  |  PropagateStatus(v), return
8  else
9  |  let X_1, ..., X_k be the possible conclusions of the rule;
10 |  if ρ ∈ {(∀′), (∀_i′)} then Y := RFmls(v)
11 |  else Y := RFmls(v) ∪ {the principal formula of ρ};
12 |  foreach 1 ≤ i ≤ k do
   |  ConToSucc(v, non-state, SType(v), null, X_i, Y, DFmls(v))
13 Status(v) := expanded;
14 foreach successor w of v with Status(w) ∉ {incomplete, unsat, sat} do
15 |  if TUnsat(w) then Status(w) := unsat
16 |  else if Type(w) = non-state then
17 |  |  v_0 := StatePred(w),  v_1 := ATPred(w);
18 |  |  if SType(v_0) = simple then
19 |  |  |  let ∃R.C be the form of CELabel(v_1);
20 |  |  |  X := {D | ∀R⁻.D ∈ Label(w) and D ∉ AFmls(v_0)}
21 |  |  else
22 |  |  |  let a:∃R.C be the form of CELabel(v_1);
23 |  |  |  X := {a:D | ∀R⁻.D ∈ Label(w) and (a:D) ∉ AFmls(v_0)}
24 |  |  if X ≠ ∅ then
25 |  |  |  if ConvMethod(v_0) = 0 then
26 |  |  |  |  FmlsRC(v_0) := FmlsRC(v_0) ∪ X;
27 |  |  |  |  if X ∩ DFmls(v_0) ≠ ∅ then Status(v_0) := unsat, return
28 |  |  |  else if X ∩ DFmls(v_0) ≠ ∅ then Status(w) := unsat
29 |  |  |  else
30 |  |  |  |  AltFmlSetsSCP(v_1) := AltFmlSetsSCP(v_1) ∪ {X};
31 |  |  |  |  Status(w) := incomplete
   |  |
32 |  else if TSat(w) then Status(w) := sat
33 UpdateStatus(v);
34 if Status(v) ∈ {incomplete, unsat, sat} then PropagateStatus(v)
```

that if the premise is satisfiable w.r.t. $\mathcal{T}$ then all of the conclusions are also satisfiable w.r.t. $\mathcal{T}$.

We write X, φ or φ, X to denote $X \cup \{\varphi\}$, and write X, Y to denote $X \cup Y$. Our *tableau calculus* $C_{\mathcal{ALCI}}$ for $\mathcal{ALCI}$ w.r.t. the TBox $\mathcal{T}$ consists of rules which

Procedure `ApplyConvRule`(v)

Global data: a rooted graph (V, E, ν).
Purpose: applying the rule $(conv)$ to the node v.

let w be the only successor of v, $E := E \setminus \{(v, w)\}$;
if $ConvMethod(w) = 0$ **then**
  $newLabel := Label(v) \cup FmlsRC(w)$;
  `ConToSucc`$(v, \mathsf{non\text{-}state}, SType(v), \mathsf{null}, newLabel, RFmls(v), DFmls(v))$
else
  let $\{\varphi_1\}, \ldots, \{\varphi_n\}$ be all the singleton sets belonging to $AltFmlSetsSC(w)$, and let $remainingSetsSC$ be the set of all the remaining sets;
  foreach $1 \leq i \leq n$ **do**
    $newLabel := Label(v) \cup \{\varphi_i\}$,
    $newDFmls := DFmls(v) \cup \{\varphi_j \mid 1 \leq j < i\}$;
    `ConToSucc`$(v, \mathsf{non\text{-}state}, SType(v), \mathsf{null}, newLabel, RFmls(v), newDFmls)$
  $Y := \{\varphi_i \mid 1 \leq i \leq n\}$;
  foreach $X \in remainingSetsSC$ **do**
    `ConToSucc`$(v, \mathsf{non\text{-}state}, SType(v), \mathsf{null}, Label(v) \cup X, RFmls(v), DFmls(v) \cup Y)$

Procedure `ApplyTransRule`(ρ, u)

Global data: a rooted graph (V, E, ν).
Purpose: applying the transitional rule ρ, which is $(\exists)$ or $(\exists')$, to the state u.

let $X_1, \ldots, X_k$ be all the conclusions of the rule ρ with $Label(u)$ as the premise;
if $\rho = (\exists)$ **then**
  let $\exists R_1.C_1, \ldots, \exists R_k.C_k$ be the corresponding principal formulas;
  foreach $1 \leq i \leq k$ **do**
    $v :=$ `NewSucc`$(u, \mathsf{non\text{-}state}, \mathsf{simple}, \exists R_i.C_i, X_i, \emptyset, \emptyset)$;
    $FmlsRC(u) := FmlsRC(u) \cup \{D \mid \forall R_i^-.D \in Label(v)$ and $D \notin$ `AFmls`$(u)\}$
else
  let $a_1 : \exists R_1.C_1, \ldots, a_k : \exists R_k.C_k$ be the corresponding principal formulas;
  foreach $1 \leq i \leq k$ **do**
    $v :=$ `NewSucc`$(u, \mathsf{non\text{-}state}, \mathsf{simple}, a_i : \exists R_i.C_i, X_i, \emptyset, \emptyset)$;
    $FmlsRC(u) := FmlsRC(u) \cup \{a_i : D \mid \forall R_i^-.D \in Label(v),\ a : D \notin$ `AFmls`$(u)\}$
if $FmlsRC(u) \cap DFmls(u) \neq \emptyset$ **then** $Status(u) := \mathsf{unsat}$;
while $Status(u) \neq \mathsf{unsat}$ *and there exists a node* w *in the local graph of* u *such that* $Status(w) = \mathsf{unexpanded}$ *and a unary rule* $\rho \neq (forming\text{-}state)$ *is applicable to* w **do** `Apply`(ρ, w);
if $Status(u) \neq \mathsf{unsat}$ **then**
  if $FmlsRC(u) \neq \emptyset$ **then** $Status(u) := \mathsf{incomplete}$
  else $ConvMethod(u) := 1$

are partially specified in Table 1 together with two special rules $(forming\text{-}state)$ and $(conv)$.

Function `Tableau`$(\mathcal{T}, \mathcal{A})$

Input: a knowledge base $(\mathcal{T}, \mathcal{A})$ in NNF in the logic $\mathcal{ALCI}$.
Global data: a rooted graph (V, E, ν).

$X := \mathcal{A} \cup \{(a:C) \mid C \in \mathcal{T}$ and a is an individual occurring in $\mathcal{A}\}$;
$\nu :=$ `NewSucc`(null, non-state, complex, null, $X, \emptyset, \emptyset$);
if `TUnsat`(ν) **then** $Status(\nu) :=$ unsat
else if `TSat`(ν) **then** $Status(\nu) :=$ sat;
while $(v :=$ `ToExpand`$()) \neq$ null **do**
    choose a tableau rule ρ different from $(conv)$ and applicable to v;
    `Apply`(ρ, v); `// defined on page` 471
return (V, E, ν)

Procedure `UpdateStatus`(v)

Global data: a rooted graph (V, E, ν).
Input: a node $v \in V$ with $Status(v) =$ expanded.

if `Kind`$(v) =$ or-node **then**
    if *some successors of* v *have status* sat **then** $Status(v) :=$ sat
    else if *all successors of* v *have status* unsat **then** $Status(v) :=$ unsat
    else if *every successor of* v *has status* incomplete *or* unsat **then**
        if v *has a successor* w *such that* $Type(w) =$ state **then**
            `//` w `is the only successor of` v
            `Apply`$((conv), v)$
        else $Status(v) :=$ incomplete
else `//` `Kind`$(v) =$ and-node
    if *all successors of* v *have status* sat **then** $Status(v) :=$ sat
    else if *some successors of* v *have status* unsat **then** $Status(v) :=$ unsat
    else if v *has a successor* w *with* $Status(w) =$ incomplete **then**
        $AltFmlSetsSC(v) := AltFmlSetsSCP(w)$, $Status(v) :=$ incomplete

Procedure `PropagateStatus`(v)

Global data: a rooted graph (V, E, ν).
Input: a node $v \in V$ with $Status(v) \in \{$incomplete, unsat, sat$\}$.

foreach *predecessor* u *of* v *with* $Status(u) =$ expanded **do**
    `UpdateStatus`(u);
    if $Status(u) \in \{$incomplete, unsat, sat$\}$ **then** `PropagateStatus`(u)

The rules $(\exists)$ and $(\exists')$ are the only "and"-rules and the only *transitional rules*. The other rules of $C_{\mathcal{ALCI}}$ are "or"-rules, which are also called *static rules*. The transitional rules are used to expand states of tableaux, while the static rules are used to expand non-states of tableaux.

For any rule of $C_{\mathcal{ALCI}}$ except $(forming\text{-}state)$ and $(conv)$, the distinguished formulas of the premise are called the *principal formulas* of the rule. The rules $(forming\text{-}state)$ and $(conv)$ have no principal formulas. As usually, we assume

that, for each rule of $C_{\mathcal{ALCI}}$ described in Table 1, the principal formulas are not members of the set X which appears in the premise of the rule.

For any state w, every predecessor v of w is always a non-state. Such a node v was expanded and connected to w by the static rule $(forming\text{-}state)$. The nodes v and w correspond to the same element of the domain of the interpretation under construction. In other words, the rule $(forming\text{-}state)$ "transforms" a non-state to a state. It guarantees that, if $\texttt{BeforeFormingState}(v)$ holds then v has exactly one successor, which is a state.

See the long version [8] for more discussions, in particular, on the use of the attribute *RFmls* and the ways of dealing with converses (i.e. with inverse roles).

The priorities of the rules of $C_{\mathcal{ALCI}}$ are as follows (the bigger, the stronger): $(\sqcap)$, $(\sqcap')$, $(\forall')$, $(\forall'_i)$: 5; $(\sqcup)$, $(\sqcup')$: 4; $(forming\text{-}state)$: 3; $(\exists)$, $(\exists')$: 2; $(conv)$: 1.

The conditions for applying a rule $\rho \neq (conv)$ to a node v are as follows:

- the rule has $Label(v)$ as the premise (thus, the rules $(\sqcap)$, $(\sqcup)$, $(\exists)$ are applicable only to simple nodes, and the rules $(\sqcap')$, $(\sqcup')$, $(\forall')$, $(\forall'_i)$, $(\exists')$ are applicable only to complex nodes)
- all the conditions accompanying with ρ in Table 1 are satisfied
- if ρ is a transitional rule then $Type(v) = \mathsf{state}$
- if ρ is a static rule then $Type(v) = \mathsf{non\text{-}state}$ and
  - if $\rho \in \{(\sqcap), (\sqcup), (\sqcap'), (\sqcup')\}$ then the principal formula of ρ does not belong to $RFmls(v)$, else if $\rho \in \{(\forall'), (\forall'_i)\}$ then the formula $b\!:\!C$ occurring in the rule does not belong to $\texttt{AFmls}(v)$
  - no static rule with a higher priority is applicable to v.

Application of a tableau rule ρ to a node v is specified by procedure $\texttt{Apply}(\rho, v)$ given on page 471. This procedure uses procedures `ApplyConvRule` and `ApplyTransRule` given on page 472. Auxiliary functions are defined on page 470. Procedures used for updating and propagating statuses of nodes are defined on page 473. The main function $\texttt{Tableau}(\mathcal{T}, \mathcal{A})$ is also defined on page 473. It returns a rooted "and-or" graph called a $C_{\mathcal{ALCI}}$*-tableau* for the knowledge base $(\mathcal{T}, \mathcal{A})$. See the long version [8] of this paper for the proof of the following theorem.

Theorem 3.1. *Let $(\mathcal{T}, \mathcal{A})$ be a knowledge base in NNF of the logic $\mathcal{ALCI}$. Then procedure* $\texttt{Tableau}(\mathcal{T}, \mathcal{A})$ *runs in exponential time (in the worst case) in the size of $(\mathcal{T}, \mathcal{A})$ and returns a rooted "and-or" graph $G = (V, E, \nu)$ such that $(\mathcal{T}, \mathcal{A})$ is satisfiable iff $Status(\nu) \neq \mathsf{unsat}$.*

4 Conclusions

We have given the first direct cut-free ExpTime (optimal) tableau decision procedure, which is not based on transformation or on the pre-completion technique, for checking satisfiability of a knowledge base in the description logic $\mathcal{ALCI}$. This satisfiability problem is more general than the problem of checking satisfiability of a concept w.r.t. a TBox studied by Goré and Widmann for $\mathcal{ALCI}$ in [4]. Our

technique is more advanced than the one of [4] as we do also global caching for nodes in the local graphs of non-states v obtained after a transition (in [4], such local graphs are trees) and we check incompatibility w.r.t. converse as soon as possible. We also delay applications of the converse rule.

Acknowledgements. This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

References

1. Ding, Y., Haarslev, V., Wu, J.: A new mapping from $\mathcal{ALCI}$ to $\mathcal{ALC}$. In: Proceedings of Description Logics (2007)
2. De Giacomo, G., Lenzerini, M.: TBox and ABox reasoning in expressive description logics. In: Aiello, L.C., Doyle, J., Shapiro, S.C. (eds.) Proceedings of KR 1996, pp. 316–327. Morgan Kaufmann, San Francisco (1996)
3. Goré, R.P., Nguyen, L.A.: EXPTIME tableaux with global caching for description logics with transitive roles, inverse roles and role hierarchies. In: Olivetti, N. (ed.) TABLEAUX 2007. LNCS (LNAI), vol. 4548, pp. 133–148. Springer, Heidelberg (2007)
4. Goré, R., Widmann, F.: Sound global state caching for ALC with inverse roles. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS, vol. 5607, pp. 205–219. Springer, Heidelberg (2009)
5. Goré, R., Widmann, F.: Optimal and cut-free tableaux for propositional dynamic logic with converse. In: Giesl, J., Hähnle, R. (eds.) IJCAR 2010. LNCS, vol. 6173, pp. 225–239. Springer, Heidelberg (2010)
6. Horrocks, I., Sattler, U.: Description logics - basics, applications, and more. In: Tutorial Given at ECAI-2002, http://www.cs.man.ac.uk/~horrocks/Slides/ecai-handout.pdf
7. Nguyen, L.A.: An efficient tableau prover using global caching for the description logic $\mathcal{ALC}$. Fundamenta Informaticae 93(1-3), 273–288 (2009)
8. Nguyen, L.A.: The long version of the current paper (2011), http://www.mimuw.edu.pl/~nguyen/alci-long.pdf
9. Nguyen, L.A., Szałas, A.: ExpTime tableaux for checking satisfiability of a knowledge base in the description logic $\mathcal{ALC}$. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS(LNAI), vol. 5796, pp. 437–448. Springer, Heidelberg (2009)
10. Nguyen, L.A., Szałas, A.: An optimal tableau decision procedure for Converse-PDL. In: Nguyen, N.-T., Bui, T.-D., Szczerbicki, E., Nguyen, N.-B. (eds.) Proceedings of KSE 2009, pp. 207–214. IEEE Computer Society, Los Alamitos (2009)
11. Nguyen, L.A., Szałas, A.: A tableau calculus for regular grammar logics with converse. In: Schmidt, R.A. (ed.) CADE-22. LNCS(LNAI), vol. 5663, pp. 421–436. Springer, Heidelberg (2009)
12. Nguyen, L.A., Szałas, A.: Checking consistency of an ABox w.r.t. global assumptions in PDL. Fundamenta Informaticae 102(1), 97–113 (2010)
13. Nguyen, L.A., Szałas, A.: Tableaux with global caching for checking satisfiability of a knowledge base in the description logic $\mathcal{SH}$. T. Computational Collective Intelligence 1, 21–38 (2010)

SWRL Rules Plan Encoding with OWL-S Composite Services

Domenico Redavid[1], Stefano Ferilli[2], and Floriana Esposito[2]

[1] Artificial Brain S.r.l., Bari, Italy
redavid@abrain.it
[2] Computer Science Department, University of Bari "Aldo Moro", Italy
{ferilli,esposito}@di.uniba.it

Abstract. This work extends a SWRL based OWL-S atomic services composition method in order to obtain and manage OWL-S composite services. After the identification of the OWL-S constructs in a SWRL plan, the steps for building the OWL-S control contructs tree, itself serializable with language syntax as well, is given. The obtained composed SWS can be considered as a Simple Process, encoded as a SWRL rule and fed to the SWRL composer for making up new compositions.

1 Introduction

The profitable use of Web services is closely tied to the automation of the following key operations: discovery, selection, composition, and invocation. The technologies developed for Web services do not allow to create intelligent tools that can automatically perform these four operations. In fact, Web services representation languages is based on XML and thus lacks a formal semantics. The Semantic Web (SW) [1] is the obvious candidate to fulfill this task. Its overlap with the technologies of Web services is referred to as Semantic Web Services (SWS) [2] whose main purpose is automating the operations mentioned above by exploiting representation languages and reasoning mechanisms developed for the SW. This relates the evolution of SWS to the results achieved in the SW. In order to exploit the advantages offered by the SW (mainly, the distributed knowledge base available on the Web), we need to use its methodologies and standards. Currently, many methods have been proposed to realize SWS composition, but almost all are based on the exploitation of methods outside SW standards. This is a problem from a practical point of view since a change of representation language is needed. Consequently it is easy to miss the semantics of information [3]. This work extends the SWRL based OWL-S[1] atomic services composition method presented in [4] in order to obtain and manage OWL-S composite services. This paper is structured as follow. In the Sec. 2 we recall the basic notions needed to understand the proposed approach. In Sec. 3 and Sec. 4 we present the logic applied to identify OWL-S constructs inside a SWRL rule plan and the applied procedure to obtain composite services. In Sec. 5 a practical example of application of our method is presented. Finally, in Sec. 6 and Sec. 7 a brief description of the state of the art about automatic approach to OWL-S composite services and a discussion on future work are given.

[1] OWL-S: Semantic markup for web services, http://www.w3.org/submission/owl-s/

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 476–482, 2011.

2 Background

In this section we briefly report the characteristics of the considered OWL-S composer and some required notions about OWL-S composite services. The work presented in [4] showed how to encode an OWL-S atomic process as a SWRL rule [5] (i.e., *inCondition* $\wedge$ *Precondition* is the body, *output* $\wedge$ *effect* is the head). After obtaining a set of SWRL rules, the following algorithm was applied: it takes as input a knowledge base containing SWRL rules set and a goal specified as a SWRL atom, and returns every possible path built combining the available SWRL rules in order to achieve such a goal. The set of paths can be considered as a SWRL rules plan (referred as *plan* in the following) representing all possible combinable OWL-S Atomic processes that lead to the intended result (the goal). The aim of this work is to describe the plan by means of an OWL-S composite process. According to OWL-S specifications, the service model defines the concept of a *Process* that describes the composition of one or more services in terms of their constituent processes. A *Process* can be *Atomic* (a description of a non-decomposable service that expects one message and returns one message in response), *Composite* (consisting of a set of processes within some control structure that defines a workflow) or *Simple* (used as an element of abstraction, i.e., a simple process may be used either to provide a view of a specialized way of using some atomic process, or a simplified representation of some composite process for purposes of planning and reasoning). As stated, OWL-S Composite processes (decomposable into other Atomic or Composite processes) can be specified by means of the following *control constructs* offered by the language: **Sequence**, **Split**, **Split + Join**, **Any-Order**, **Choice**, **If-Then-Else**, **Iterate**, **Repeat-While and Repeat-Until**, and **AsProcess**. One crucial feature of a composite process is the specification of how its inputs are accepted by particular sub-processes, and how its various outputs are produced by particular sub-processes. Structures to specify the *Data Flow* and the *Variable Bindings* are needed. When defining processes using OWL-S, there are many places where the input to one process component is obtained as one of the outputs of a preceding step, short-circuiting the normal transmission of data from service to client and back. For every different type of *Data Flow* a particular *Variable Bindings* is given. Formally, two complementary conventions to specify *Data Flow* have been identified:**consumer-pull** (the source of a datum is specified at the point where it is used) and **producer-push** (the source of a datum is managed by a pseudo-step called *Produce*). Finally, we remark that a composite process can be considered as an atomic one using the OWL-S Simple process declaration. This allows to treat Composite services during the application of the SWRL Composer.

3 Encoding the SWRL Plan with OWL-S Constructs

According to the OWL-S specification about a composed process and its syntax, it is possible to represent the composition of atomic services obtained through the SWRL rule composer by means of an OWL-S composite service. In this section we will

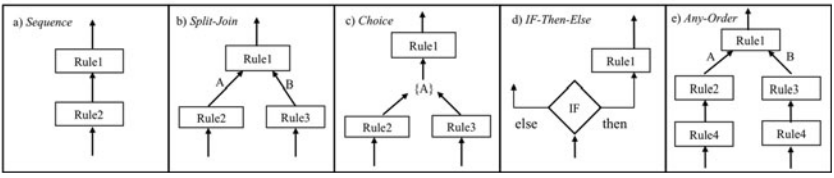


Fig. 1. The OWL-S control constructs in the plan

Table 1. OWL-S *Control Constructs* identified in the plan

Sequence. Represents the simplest situation of the plan, in which the rules are geared to each other in a sequential manner, that is the head of a rule corresponds to one of the atoms in the body of another. This indicates a sequential execution, and thus will be used the Sequence construct. According to the specification, *Sequence* is a construct whose components are executed in a given order and the result of the last element is used as the result of the whole sequence (Fig. 1 a)).
Split-Join. Represents the situation where two or more rules, with different head atoms are grafted directly into two or more atoms in the body of a particular rule. In this circumstance there is a branch that is evaluated and coded with a construct of the type Split-Join. According to the specifications, Split-Join is a construct whose components are executed simultaneously, i.e., they run competitively and with a certain level of synchronization. This construct is used because it is necessary that all grafted rules are performed successfully. The condition behind the utilization of this construct assumes that its components can be overlapped in the execution, i.e., they are all different (Fig. 1 b)).
Choice. Represents the situation where two or more rules, with the same head atoms, are grafted directly into one of the atoms in the body of a particular rule. In this circumstance there is a branch that is evaluated and encoded with the construct Choice. According to the specifications, Choice is a construct whose components are part of a set from which any one can be called for execution. This construct is used because the results from the rules set can be easily overlapped, no matter which component is going to be run because the results are always of the same type (Fig. 1 c)).
If-Then-Else. It could represent the situation where the body of a rule are the atoms that identify a precondition. In this case, the service that identifies the rule to be properly executed needs that its precondition must be true. In this circumstance, therefore, the precondition was extrapolated and used as a condition in the If-Then-Else construct. According to the specifications, If-Then-Else construct is divided into three parts: the 'then' part, the 'else' part and the 'condition' part. The semantics behind this construct is to be understood as: "if the 'condition' is satisfied, then run the 'then' part, otherwise run the 'else' part.". In this case it is understood that if the condition is satisfied then one can run the service, otherwise the service will not be executed (Fig. 1 d)).
Any-Order. Represents a situation similar to the Split-Join, but this particular case covers those circumstances where control constructs or processes are present multiple times in the structure of the plan, and it is important that their execution is not overlapped in order to prevent a break in the composite process. This type of situation can be resolved through the use of the Any-Order construct because its components are all performed in a certain order but never competitively (Fig. 1 e)).

analyze how it is possibe to get one. An OWL-S composed process can be considered as a tree whose nonterminal nodes are labeled with control constructs, each of which has children that are specified through the OWL property *components*. The leaves of the tree are invocations of the processes mentioned as instances of the OWL class Perform, a class that refers to the process to be performed. Bearing in mind the characteristics of the plan constructed by means of the method specified in [4], we identify the OWL-S control constructs to be used to implement the plan applying the guidelines reported in Table 1. Currently, the OWL-S specification does not completely specify the Iteration and its dependent constructs (Repeat-While and Repeat- Until) and how the asProcess construct could be used. For this reason they are not treated in this paper, but they will be considered in future work. Furthermore, The Split construct is not covered in this discussion because for atomic services occurs only in the presence of the same input parameter.

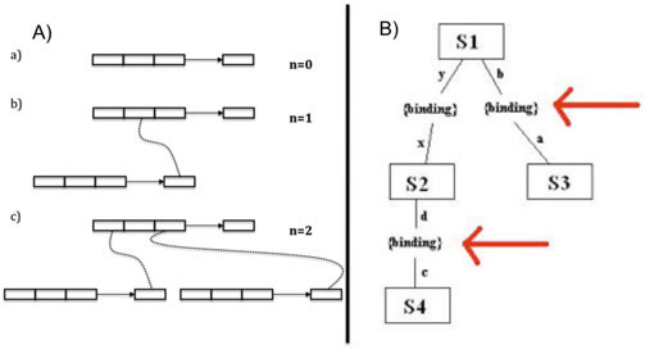


Fig. 2. A) The different types of grafted SWRL rules in the plan. B) Consumer-pull data flow.

4 The Procedure to Build the OWL-S Constructs Tree

For our purposes, each rule is represented in the composer as an object that has various features and information that could be helpful to the composition. Such an object, called *Rulebean*, contains information about:

- The atoms in the declared precondition of the rule;
- The URI of the atomic process to which the rule refers;
- The number of atoms in which the grafted rules have correspondence;
- A list containing pointers to the other Rulebeans with which it is linked.

The information about the atoms of the preconditions allow to check the presence of IF conditions that could lead to identify a situation that needs an IF-Then-Else construct. The URI of the atomic process to which the rule refers is necessary because the leaves of the constructs tree must instantiate the processes to be performed. Finally, since each rule can be linked to other rules, it is necessary to store both their quantity and a pointer to the concatenated rules. In this way each Rulebean carries within it the entire underlying structure. This structure is implemented as a tree of lists, where each list contains the Rulebeans that are grafted on the same atom. Now let's show in detail the steps needed to encode a plan with the OWL-S control constructs. Referring to Figure 2A), the procedure involves the application of three subsequent steps depending on the number of grafted rules (n):

1. Search all Rulebeans grafted with a number of rules equal to zero ($n = 0$) (Figure 2a)).
   a. Store therein an object that represents the "leaf", i.e., an executable process.
2. Search all Rulebeans grafted with a number of rules equal to one ($n = 1$) (Figure 2b).
   a. Check the engaged list:
      i. If there is only one Rulebean, the node will be of type "Sequence";
      ii. If there are multiple Rulebeans, the node will be of type "Choice";
   b. Store the object representing the created structure in the Rulebean.
3. Search all Rulebeans grafted with a number of rules greater than one ($n \geq 2$) (Figure 2c).
   a. For each grafted list follow the steps in 2.a;
   b. Make the following checks on the structure:
      i. If there are repeated Rulebeans add a node of type "Any-Order";
      ii. If there are no repeated Rulebeans add a node of type "Split-Join";
   c. Stores the object representing the created structure in the Rulebean

Since the If-Then-Else construct overlaps with the constructs assigned during this procedure, it is identified in another way. During the creation of a Rulebean, a check is performed to verify if there are atoms in the body of the rule labeled as belonging to a precondition. If this is the case, the Rulebean will be identified as the "Then" part of the construct, and the atoms of the precondition will form the 'If' condition. The "Else" part will be identified as the complementary path, if existing (for the "Sequence" construct it does not exist, of course). Finally, the data flow is implemented in accordance with the consumer-pull method, i.e., the binding of variables is held exactly at the point in which it occurs. Referring to the example in Figure 2B), we can see that parameter y of $S1$ is in correspondence with parameter x of $S2$ which means that the output of $S2$ is the same, semantically speaking, as the input of $S1$. In practice, an OWL-S declaration will be created, specifying that the input y of $S1$ comes from the output x of $S2$.

Table 2. OWL-S Atomic services test set

OWL-S SERVICE NAME	INPUTS	OUTPUTS
Service-10	Books:Title	Books:Book
Service-12	Books:Title	Books:Book
Service-15	Books:Book	Books:Person
Service-28	Books:Novel	Books:Person
Service-9	Books:Person; Books:Book	Concept:Price

5 Example

Let us consider the subset of atomic services in Table 2 chosen from the OWLS-TC dataset[2]. By means of the OWL-S service composer presented in [4], we obtain the plan in Figure 3 a). Now, applying the algorithm described in Section 3, we obtain the OWL-S constructs tree of Figure 3 b). In practice, we start searching Rulebeans grafted with a number of rules equal to zero, finding Service-10, Service-12 and Service-28. These will be tree leafs. We go on searching Rulebeans grafted with a number of rules equal to one, finding Service-15. It has two rulebeans grafted on the same Atom, i.e., $books : Book$, thus we use a "Choice" construct. To link the obtained structure with Service-15 (another tree leaf) we use the "Sequence" construct, naming this structure $C1$. We continue serching Rulebeans grafted with a number of rules greater than one, finding Service-9. It has Service-10 and Service-12 grafted on the Atom $books : Book$, and Service-28 (another tree leaf) and $C1$ grafted on the Atom $books : Person$. Both pairs are linked with a "Choice" construct, and we call them $C2$ and $C3$, respectively. Since $C2$ and $C3$ contains repeated Rulebean (Service-10 in "Choice" with Service-12), we model this situation with the "Any-Order" construct. The depicted "If-Then-Else" construct is obtained applying the following consideration. Supposing that the precondition of Service-28 states that in order to execute the service the input must be exclusively a "books:Novel" (an OWL subclass of "books:Book"). Then, we can put this assertion as the IF condition, the execution of the service as the "Then" part, and the non execution of the service as the "Else" part.

[2] OWL-S service retrieval test collection, http://projects.semwebcentral.org/projects/owls-tc/

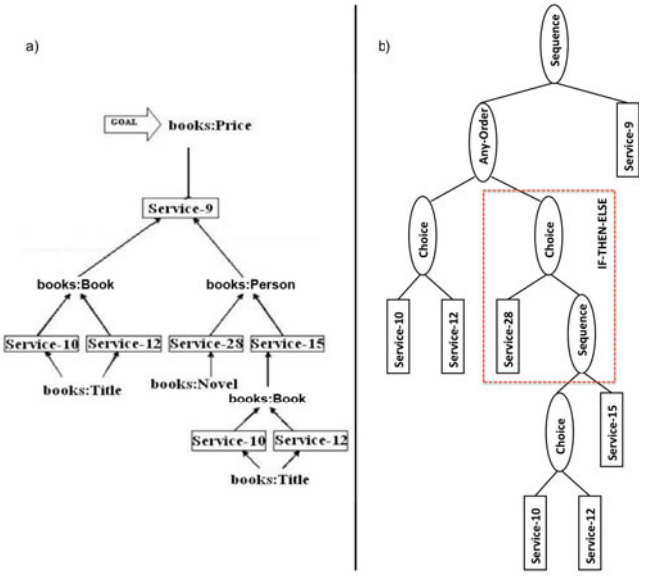


Fig. 3. The SWRL plan obtained applying the composer over the services of Table 2

6 Related Work

In this section we analyze some works on SWS composition that involve OWL-S *composite* services. In [6], the semantics underlying a relevant subset of OWL-S has been translated into First Order Logic (FOL), obtaining a set of axioms for describing the features of each service. By combining these axioms within a Petri Net, they obtain process-based service models that enable reasoning about interactions among the processes. [7] demonstrate how an OWL reasoner can be integrated within the SHOP2 AI planner for SWS composition. The reasoner is used to store the world states, answer the planners queries regarding the evaluation of preconditions, and update the state when the planner simulates the effects of services. [8] propose CASheW-S, a compositional semantic system based on Pi-Calculus in order to realize OWL-S service composition. OWL-S is encoded with the proposed system. [9] proposes CCTR, an extension of Concurrent Transaction Logic (CTR), as a formalism for modelling, verification and scheduling composite SWS. It describe how OWL-S and CCTR can be used together for modeling a complex service and constraints, and make reasoning. Clearly, all the mentioned works involve a representation change from OWL-S, which is based on description logic, to other formalisms. The disadvantage of these changes is that a loss of semantics might take place during the transformation process [3]. Furthermore, reasoners built for these formalisms work with the CWA, while SW reasoners with OWA. Conversely, our method works using only standards and tools proposed for the Semantic Web.

7 Conclusions and Future Work

The automatic SWS composition is the more complex process to achieve with only the tools built on the Description Logics [10]. In this article we presented the extension of a composition method for Semantic Web Services in order to integrate OWL-S composite services. In particular, we have shown that having a plan of atomic services represented using SWRL rules is possible to identify the corresponding OWL-S control constructs. The constructs identified in this way are used to build the OWL-S control construct tree that is directly serializable using the syntax of the language. The constructed OWL-S composite service can be considered as a Simple Process, then encoded as a SWRL rule and finally fed to the SWRL composer for obtaining new compositions. As a future work, it is important to find ways to manage the remaining constructs (Iteration) and improve the composer in order to reuse internal parts of composite process during the composition. Another objective is to integrate into the composer ontology alignment methods in order to enable Semantic interoperability. As in previous work, the emphasis will be placed on the exclusive use of technologies developed for the Semantic Web.

References

[1] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284, 34–43 (2001)
[2] McIlraith, S.A., Son, T.C., Zeng, H.: Semantic Web Services. IEEE Intelligent Systems 16, 46–53 (2001)
[3] Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artificial Intelligence 82, 353–367 (1996)
[4] Redavid, D., Iannone, L., Payne, T.R., Semeraro, G.: OWL-S Atomic Services Composition with SWRL Rules. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 605–611. Springer, Heidelberg (2008)
[5] Horrocks, I., Patel-Schneider, P.F., Bechhofer, S., Tsarkov, D.: OWL rules: A proposal and prototype implementation. J. of Web Semantics 3, 23–40 (2005)
[6] Narayanan, S., McIlraith, S.A.: Simulation, verification and automated composition of web services. In: WWW 2002: Proceedings of the 11th International Conference on World Wide Web, pp. 77–88. ACM Press, New York (2002)
[7] Sirin, E., Parsia, B.: Planning for Semantic Web Services. In: Semantic Web Services Workshop at 3rd International Semantic Web Conference (2004)
[8] Norton, B., Foster, S., Hughes, A.: A compositional operational semantics for owl-s. In: Bravetti, M., Kloul, L., Tennenholtz, M. (eds.) EPEW/WS-EM 2005. LNCS, vol. 3670, pp. 303–317. Springer, Heidelberg (2005)
[9] Senkul, P.: Modeling composite web services by using a logic-based language. In: Bouquet, P., Tummarello, G. (eds.) SWAP. CEUR Workshop Proceedings, CEUR-WS.org, vol. 166 (2005)
[10] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2003)

Detecting Web Crawlers from Web Server Access Logs with Data Mining Classifiers

Dusan Stevanovic, Aijun An, and Natalija Vlajic

Department of Computer Science and Engineering, York University,
4700 Keele Street, Toronto, Ontario, Canada
{dusan,aan,vlajic}@cse.yorku.ca

Abstract. In this study, we introduce two novel features: the consecutive sequential request ratio and standard deviation of page request depth, for improving the accuracy of malicious and non-malicious web crawler classification from static web server access logs with traditional data mining classifiers. In the first experiment we evaluate the new features on the classification of known well-behaved web crawlers and human visitors. In the second experiment we evaluate the new features on the classification of malicious web crawlers, unknown visitors, well-behaved crawlers and human visitors. The classification performance is evaluated in terms of classification accuracy, and F_1 score. The experimental results demonstrate the potential of the two new features to improve the accuracy of data mining classifiers in identifying malicious and well-behaved web crawler sessions.

Keywords: Web Crawler Detection, Web Server Access Logs, Data Mining, Classification, DDoS, WEKA.

1 Introduction

Today, the world is highly dependent on the Internet, the main infrastructure of the global information society. Consequently, the availability of Internet is very critical for the economic growth of the society. For instance, the way traditional essential services such as banking, transportation, medicine, education and defence are operated is now actively replaced by cheaper and more efficient Internet-based applications. However, the inherent vulnerabilities of the Internet architecture provide opportunities for various attacks on its security. Distributed denial-of-service (DDoS) is an example of a security attack with particularly severe effect on the availability of the Internet. United States' Department of Defence report from 2008, presented in [1], indicates that cyber attacks in general (and DDoS attacks in particular) from individuals and countries targeting economic, political, and military organizations may increase in the future and cost billions of dollars.

An emerging (and increasingly more prevalent) types of DDoS attacks - known as Application Layer or Layer-7 attacks - are shown to be especially challenging to detect. Namely, the traditional network measurement systems are shown to be rather ineffective in identifying the presence of Layer-7 DDoS attacks. The reason for this is that in an application layer attack, in order to cripple or completely disable a Web server, the

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 483–489, 2011.

attacker utilizes legitimate-looking network sessions. For instance, HTML requests sent to a web server may be cleverly constructed to perform semi-random walks of web site links. Since the attack signature resembles legitimate traffic, it is difficult to construct an effective metric to detect and defend against the Layer-7 attacks.

So far, a number of studies on the topic of application-layer DDoS attacks have been reported. Thematically, these studies can be grouped into two main categories: 1) detection of application-layer DDoS attacks during a *flash crowd* event based on aggregate-traffic analysis and 2) differentiation between well-behaved and malicious web crawlers based on web-log analysis.

In this study, we introduce two new features, in addition to traditional features, to improve the detection of malicious and non-malicious web crawlers with data mining classifiers. Namely, we performed two sets of experiments. The goal of the first experiment is to evaluate whether the two new features improve the detection of web crawlers from sessions of human visitors and well-behaved web crawlers. The goal of the second experiment is to evaluate whether the two new features improve the detection of malicious crawlers and unknown web visitors from sessions of human visitors, well-behaved crawlers, malicious crawlers and unknown visitors (either human or robot). The implementations of classification algorithms, utilized in our study, are provided by WEKA data mining software [2].

The paper is organized as follows: In Section 2, we discuss previous works on web crawler detection. In Section 3, we present an overview of the web crawler classification by employing a log analyzer pre-processor. In Section 4, we outline the design of the experiments and the performance metrics that were utilized. In Section 5, we present and discuss the results obtained from the classification study. In Section 6, we conclude the paper with our final remarks.

2 Related Work

In over the last decade, there have been numerous studies that have tried to classify web robots from web server access logs. One of the first studies on classification of web robots using data mining classification techniques is presented in [3]. In the study, authors attempt to discover web robot sessions by utilizing a feature vector of properties of user sessions. This classification model when applied to a dataset suggests that robots can be detected with more than 90% accuracy after only four issued web-page requests. Other techniques have also been proposed that utilize similar classification methods, namely, [4] and [5].

The novelty of our research is twofold. Firstly, to the best of our knowledge, this is the first study that classifies web crawlers as malicious, known well-behaved web robots (such as Googlebot and MSNbot among others) and unknown visitors (potential crawlers). Secondly, in addition to employing traditional features in our classification, we also introduce two new features and evaluate whether the utilization of these additional features can improve the classification accuracy rates.

3 Dataset Preparation

Crawlers are programs that traverse the Internet autonomously, starting from a *seed* list of web pages and recursively visiting documents accessible from that list. Their

primary purpose is to discover and retrieve content and knowledge from the Web on behalf of various Web-based systems and services. In this section we describe how crawlers can be detected by a simple pattern matching pre-processor with log analyzer functionality. The pre-processing task consists of identifying sessions, extracting features of each session and lastly performing session classification.

3.1 Session Identification

Session identification is the task of dividing a server access log into sessions. Typically, session identification is performed by first grouping all HTTP requests that originate from the same user-agent, and second by applying a timeout approach to break this grouping into different sub-groups, so that the time-lapse between two consecutive sub-groups is longer than a pre-defined threshold. Usually, a 30-min period is adopted as the threshold in Web-mining studies [3].

3.2 Features

From previous web crawler classification studies, namely [3], [4], and [5], we have adopted seven different features that are shown to be useful in distinguishing between browsing patterns of web robots and humans. These features are: 1) click rate, 2) HTML-to-Image Ratio, 3) Percentage of PDF/PS file requests, 4) Percentage of 4xx error responses, 5) Percentage of HTTP requests of type HEAD, 6) Percentage of requests with unassigned referrers and 7) 'Robot.txt' file request. (Note that in the rest of the paper we will refer to these features based on their numeric ID shown here).

As mentioned earlier, features 1-7 have been used in the past for distinguishing between human- and robot-initiating sessions. However, based on the recommendations and discussion presented in [6], we have derived two additional and novel features in web robot classification:

8. Standard deviation of requested page's depth – a *numerical* attribute calculated as the standard deviation of page depth across all requests sent in a single session. For instance, we assign a depth of three to a web page '/cshome/courses/index.html' and a depth of two to a web page '/cshome/calendar.html'. This attribute should be low for web robot sessions and high for human sessions since a web robot should scan over a narrower directory structure of a web site than a human user.
9. Percentage of consecutive sequential HTTP requests – a *numerical* attribute calculated as the number of sequential requests for pages belonging to the same web directory and generated during a single user session. For instance, a series of requests for web pages matching pattern '/cshome/course/*.* will be marked as consecutive sequential HTTP requests. However, a request to web page '/cshome/index.html' followed by a request to a web page 'cshome/courses/index.html' will not be marked as consecutive sequential requests. This attribute should be high for human user sessions since most Web browsers, for example, retrieve the HTML page, parse through it, and then automatically send a barrage of requests to the server for embedded resources on the page such as images, videos, and client side scripts to execute. Alternatively, it should be low for crawler-based sessions because web robots are able to make their own decisions on what resource should be requested from the web server.

3.3 Dataset Labelling

The sessions of the training/testing dataset are classified as belonging to a certain class based on the user agent field found in the users' requests. The log analyzer maintains a table of user agent fields of all known (malicious or well-behaved) web crawlers as well as of all known browsers. (This table can be built from the data found on web sites in [7] and [8]). Also note that sessions with the name of known browsers in the user agent field are identified/labelled as belonging to a human visitor.) Out of all identified user sessions, 70% of sessions are placed in the training dataset while the rest are placed in the test dataset. This exclusive separation of user sessions ensures a fair investigation since, following the training phase, the classifiers are tested on previously unseen sessions.

In this study, we perform two types of classifications/experiments. In the first experiment we examine whether human users (class label = 0) and well-behaved crawlers (class label = 1) can be separated by the classification algorithms. In the second experiment we examine whether human users and well-behaved crawlers (class label = 0) can be separated from known malicious crawlers and unknown visitors (class label = 1) by the classification algorithms.

4 Experimental Design

4.1 Web Server Access Logs

The datasets were constructed by pre-processing web server access log files provided by York CSE department. The log file stores detailed information about user web-based access into the domain www.cse.yorku.ca during a 4-week interval - between mid December 2010 and mid January 2011. There are a total of about 3 million log entries in the file. Tables 1 list the number of sessions and class label distributions generated by the log analyzer for experiments 1 and 2.

Table 1. Class distribution in training and testing datasets used in Experiment #1 and # 2

	Training	Testing
Total Number of Sessions	96845	36789
Total # of Session with Class Label = 0 in Experiment 1	94723	35933
Total # of Session with Class Label = 1 in Experiment 1	2122	856
Total # of Session with Class Label = 0 in Experiment 2	94263	35864
Total # of Session with Class Label = 1 in Experiment 2	2582	925

4.2 Classification Algorithms

The detection of web crawlers was evaluated with the following seven classifiers: C4.5 [9], RIPPER [10], Naïve Bayesian, Bayesian Network, k-Nearest Neighbour (with k=1), LibSVM and Multilayer Perceptron (MP), a neural network algorithm. The implementation of each algorithm is provided in the WEKA software package (more on data mining classification algorithms can be found in [11]). Each classifier is trained on one training dataset and then tested on another supplementary dataset. In the testing phase, in order to determine the classification accuracy, the classification

results generated by the classifiers are compared against the 'correct' classifications derived by the log analyzer.

4.3 Performance Metrics

In order to test the effectiveness of our classifiers, we adopted metrics that are commonly applied to imbalanced datasets: recall, precision, and the F_1-score [5], which summarizes both recall and precision by taking their harmonic mean. F_1 score summarizes the two metrics into a single value, in a way that both metrics are given equal importance. The F_1-score penalizes a classifier that gives high recall but sacrifices precision and vice versa. For example, a classifier that classifies all examples as positive has perfect recall but very poor precision. Recall and precision should therefore be close to each other, otherwise the F_1-score yields a value closer to the smaller of the two.

5 Classification Results

The motivation for the first experiment was to evaluate whether features 8 and 9 can improve the accuracy in classifying sessions as either belonging to a human user or a well-behaved web crawler. The motivation for the second experiment was to evaluate whether features 8 and 9 can improve the accuracy in classifying sessions as belonging to malicious web crawlers and unknown visitors.

5.1 Experiment 1 Results

The Figure 1 shows the difference in terms of percentage points between the accuracy rates of classification results when all 9 features and when only features 1-7 are utilized. As can be observed, there is a slight improvement in accuracy rate when all 9

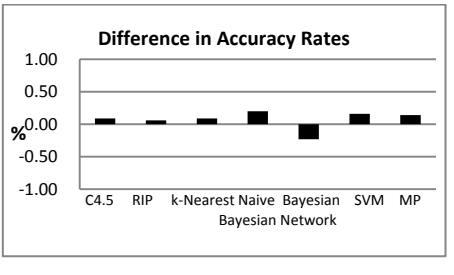


Fig. 1. Difference in terms of percentage points between the accuracy rates of classification results when all 9 features and when only features 1-7 are utilized.

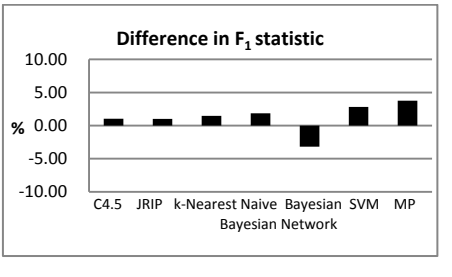


Fig. 2. Difference between F_1 score for the seven algorithms that are trained on the data set containing all 9 features and F_1 score for the seven algorithms that are trained on the data set containing only first 7 features

features are used for all algorithms except the Bayesian Network which shows a slight decline in the accuracy rate. The actual classification accuracies when features 1-7 are applied were already very high, above 95% in the case of all seven classifiers. This explains such a modest improvement in classification accuracy when all 9 features are applied.

A more accurate evaluation of classifiers' performance can be derived by examining the recall, precision and F_1 score metrics. Figure 2 shows the difference in terms of percentage points between the F_1 score for the seven algorithms that are trained on the data set containing all 9 features and F_1 score for the seven algorithms that are trained on the data set containing only first 7 features. As can be observed, the use of 9 features results in noticeably higher F_1 score (between 0.5% up to nearly 4%) in six out of seven examined algorithms.

5.2 Experiment 2 Results

The Figure 3 shows the difference in terms of percentage points between the accuracy rates of classification results when all 9 features and when only features 1-7 are utilized. As can be observed, there is a slight improvement in accuracy rate when all 9 features are used for all algorithms except in the scenario where SVM algorithm is employed which shows a slight decline in accuracy. The modest improvement in classification accuracies is again due to high accuracies (from 95% to 99%) achieved by the classifiers when only 7 features were used.

A more accurate evaluation of classifiers' performance can be made by applying the recall, precision and F_1 score metrics. Figure 4 shows the difference in terms of percentage points between the F_1 score for the seven algorithms that are trained on the data set containing all 9 features and F_1 score for the seven algorithms that are trained on the data set containing only first 7 features. As can be observed, in almost all seven algorithms (except for SVM) the F_1 score is noticeably higher (between 0.5% up to nearly 7%) when all 9 features are used to train the classification algorithms.

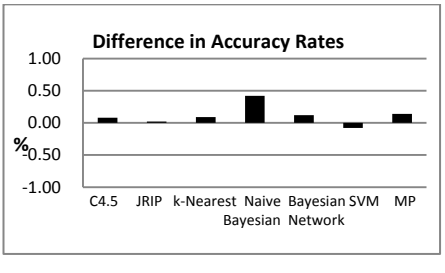


Fig. 3. Difference in terms of percentage points between the accuracy rates of classification results when all 9 features and when only features 1-7 are utilized.

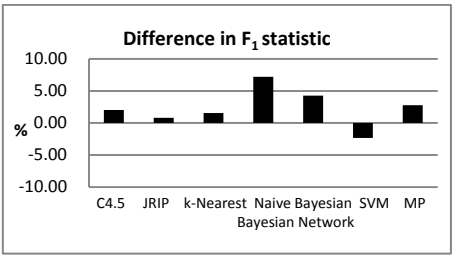


Fig. 4. Difference between F_1 score for the seven algorithms that are trained on the data set containing all 9 features and F_1 score for the seven algorithms that are trained on the data set containing only first 7 features

6 Conclusion and Final Remarks

In this paper, we propose two new features for detecting known well-behaved web crawlers, known malicious web crawlers, unknown and human visitors of a university web site using existing data mining classification algorithms. Additionally, this is the first study that attempts to classifiers both non-malicious and malicious crawlers from web server access logs.

The results conclusively show that the two new features proposed, the consecutive sequential requests ratio and standard deviation of page request depths, improve the classification accuracy and F_1 score when most of the classification algorithms are employed.

As evident in our study, the browsing behaviours of web crawlers (both malicious and well-behaved) and human users are significantly different. Therefore, from the data mining perspective, their identification/classification is very much a feasible task. However, the identification/classification of malicious crawlers that attempt to mimic human users will remain the most difficult future classification challenge.

References

1. Wilson, C.: Botnets, Cybercrime, and Cyberterrorism: Vulnerabilities and Policy Issues for Congress. Foreign Affairs, Defense, and Trade Division, United States Government (2008)
2. WEKA (December 2010), http://www.cs.waikato.ac.nz/ml/weka/
3. Tan, P.N., Kumar, V.: Patterns, Discovery of Web Robot Sessions Based on their Navigation. Data Mining and Knowledge Discovery 6, 9–35 (2002)
4. Bomhardt, C., Gaul, W., Schmidt-Thieme, L.: Web Robot Detection - Preprocessing Web Logfiles for Robot Detection. In: Proc. SISCLADAG, Bologna, Italy (2005)
5. Stassopoulou, A., Dikaiakos, M.D.: Web robot detection: A probabilistic reasoning approach. Computer Networks: The International Journal of Computer and Telecommunications Networking 53, 265–278 (2009)
6. Doran, D., Gokhale, S.S.: Web robot detection techniques: overview and limitations. Data Mining and Knowledge Discovery, 1–28 (June 2010)
7. User-Agents.org (January 2011), http://www.user-agents.org
8. Bots vs. Browsers (January 2011), http://www.botsvsbrowsers.com/
9. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
10. Cohen, W.W.: Fast effective rule induction. In: ICML 1995, pp. 115–123 (1995)
11. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, San Francisco (2006)

To Diversify or Not to Diversify Entity Summaries on RDF Knowledge Graphs?

Marcin Sydow[1,2], Mariusz Pikuła[2], and Ralf Schenkel[3]

[1] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
[2] Polish-Japanese Institute of Information Technology, Warsaw, Poland
[3] Saarland University and MPI for Informatics, Saarbrücken, Germany
{msyd,mariusz.pikula}@poljap.edu.pl,
schenkel@mmci.uni-saarland.de

Abstract. This paper concerns the issue of *diversity* in entity summarisation on RDF knowledge graphs. In particular, we study whether and to what extent the notion of diversity is appreciated by real users of a summarisation tool. To this end, we design a user evaluation study and experimentally evaluate and compare, on real data concerning the movie domain (IMDB), two graph-entity summarisation algorithms: PRECIS and DIVERSUM, that were proposed in our recent work. We present successful experimental results showing that diversity-awareness of a graph entity summarisation tool is a valuable feature and that DIVERSUM algorithm receives quite positive user feedback.

Keywords: diversity, entity summarisation, RDF graphs, experiments, user evaluation.

1 Introduction

Semantic knowledge representation in the form of RDF-style graphs is gaining importance in research and applications especially in the context of automatic knowledge harvesting from open-domain sources such as WWW.

In such knowledge graphs the nodes represent entities (e.g. in the movie domain: actors, directors, or movies) and the directed arcs represent relations between entities (e.g. "acted in", "directed", "has won prize"). Equivalently, each arc of such a graph, together with its ends can be called a "RDF triple" and corresponds to subject and object (nodes) connected by a predicate (arc label). SPARQL[1] query language, or alike, can be used to query such knowledge graphs. While it is powerful and expressive, it demands that the user knows its sophisticated syntax and, in addition, has some pre-requisite knowledge of the base being queried (e.g. names of relations, etc.). Thus, if one wants to make querying RDF-like knowledge bases accessible to broad audience, there is a need to "relax" querying syntax to make it easy for ordinary users.

Probably the simplest imaginable "query" would be like "tell me something about an entity x", i.e., to ask for a *summary* of important facts about an entity x specified by the user, together with a small limit on the number of selected facts (triples). The main

[1] http://www.w3.org/TR/rdf-sparql-query/

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 490–500, 2011.

issue would be how to *select* a small number of facts to include into the summary to make it useful. Additionally we focus here on the issue of *diversity* of such a summary.

Summarisation has been intensively studied for text documents, for example by Wan et al. [12]or, for multiple documents, by Wan [11]. Zhang et al. [13] (as a recent example for a large set of similar papers) consider the problem of creating concise summaries for very large graphs such as social networks or citation graphs; in contrast to this, our work aims at summarising information around a single node in a graph. Ramanath et al. [5,6] propose methods for summarising tree-structured XML documents within a constrained budget. The problem of entity summarisation in the form of a graph with limited number of edges was originally proposed in [8], together with an efficient algorithm, called PRECIS.

The concept of diversity has recently attracted much interest in IR research (over 50 works at the time of writing). Due to space limitations we refer only to selected works here. In particular, the algorithms studied in this paper are indirectly derived from the concepts presented in [3,1] in the ways that are described in [7]. The issue of results diversification in structured data, e.g. in relational databases was studied in [10].

The problem of entity summarisation on RDF graphs has been recently proposed and studied in [8] together with a PRECIS algorithm, and in [7] where a *diversified* variant of the problem, named DIVERSUM, was introduced.

This paper builds on our previous work in [8,7]. Contributions of this paper are as follows: 1) synthesis of our previous work on entity summarisation with limited edge budget on RDF knowledge graphs; 2) motivation, based on recent advances in information retrieval, for diversification of summaries; 3) design, implementation and running a user evaluation experiment assessing the quality of both PRECIS and DIVERSUM algorithms and comparing them; 3) report and detailed analysis of experimental results on a real data set from the movie domain (IMDB) that confirms both the need for diversity and good absolute result quality of the DIVERSUM algorithm.

2 Problem Formulation and Baseline Algorithm (PRECIS)

We define the problem of entity summarisation on knowledge graphs as follows:

INPUT: 1) G – an underlying knowledge base (a multi-digraph with positive real weights on arcs representing triples' "importance"); 2) q – a node of G (entity to be summarised); 3) $k \in N$ – a limit on edges (triples) in the summary

OUTPUT: S – a connected subgraph of G of at most k arcs, containing q , being a "good summary" of q

In [8] an efficient greedy algorithm for the above problem, called PRECIS, was introduced and preliminarily evaluated. The idea of the algorithm is simple, since it constructs the summary by greedily selecting the edges that are closest (according to the weights on arcs) to the input node (entity) until k edges are selected. The algorithm could be viewed as the adaptation of the Dijkstra's shortest paths algorithm to multigraphs, arc number constraint, and to focusing on arcs rather than nodes (see figure 1).

Since the weights on arcs represent "importance" of triples (facts), the output of the algorithm could be regarded as the selection of the top-k most important facts concerning the summarised entity. One should notice here the analogy to the classic PRP

```
PriorityQueue PQ; Set RESULT;
forEach a in radius k from q: a.distance := "infinity"
forEach a adjacent to q: {a.distance := a.weight; PQ.insert(a)}

while( (RESULT.size < k) and ((currentArc = PQ.delMin()) != null) )
  forEach a in currentArc.adjacentArcs:
    if (not RESULT.contains(a)) then
      a.distance := min(a.distance, (a.weight + currentArc.distance))
      if (not PQ.contains(a)) then PQ.insert(a)
      else PQ.decreaseKey(a,a.distance)
  RESULT.add(currentArc)
return RESULT
```

Fig. 1. The PRECIS algorithm for computing entity summarisation. The algorithm is an adaptation of the Dijkstra's single-source shortest paths algorithm. Each arc a has two real attributes: $weight$ and $distance$ as well as an $adjacentArcs$ attribute that keeps the set of arcs sharing a node with a (except a itself). PQ is a min-type priority queue for keeping the arcs being processed, with the value of weight serving as the priority, and $RESULT$ is a set. PQ and $RESULT$ are initially empty. We also assume that "infinity" is a special numeric value being greater than any real number.

("Probability Ranking Principle") model, formerly used in information retrieval, where top-k most relevant documents are selected as the result for a search query q. PRP model has been criticised in IR community for producing results that can be redundant and dominated by the most popular interpretation of a query (when it is ambiguous).

Similarly, the PRECIS algorithm that focuses solely on "importance" of facts has high risk of producing results that are redundant in terms of relation type of selected facts. Example of the PRECIS's output with this problem is shown on figure 3 (left).

3 Diversified Entity Summarisation on RDF-Graphs (DIVERSUM)

To improve the results of importance-based PRECIS summarisation algorithm described in section 2 we follow the analogy with information retrieval, where many ideas on search result diversification were proposed to avoid redundancy.

An example of an early work in IR in this spirit is [2], where the result optimises a balanced combination of "relevance" and "novelty" in a model called MMR ("Maximal Marginal Relevance"). This idea has been further intensively studied and extended in numerous recent papers (see [3,1], for example).

Adapting the general ideas on diversification from IR to the case of knowledge graphs, [7] proposed a problem of DIVERSUM that concerns producing a *diversified* entity summarisation with limited edge budget on RDF knowledge graphs together with an algorithm that is a modification of PRECIS. Compared to PRECIS, that focuses solely on "importance" of selected facts, the DIVERSUM algorithm[2] proposed in [7] applies a modified fact selection procedure so that it focuses not only on "importance"

[2] We will use the term DIVERSUM both for the algorithm and the name of the problem.

but also on "novelty" and "popularity" of selected facts. In short, the algorithm greedily selects triples from those connected to the current result (initially consisting solely of the summarised entity), and the priority is determined by (first) *novelty* (arc label not present in the result yet), (second) *popularity* (arc multiplicity) and (third) *importance* (arc weight) (see figure 2 for a pseudo-code).

1. $dist = 1$; $S = \emptyset$; $\Lambda = \emptyset$;
2. `while` in $zone(dist, q, S)$ there is still an arc with the label that is not in Λ:
   (a) select a highest-multiplicity label $l \notin \Lambda$ in $zone(dist, q, S)$; $\Lambda.add(l)$
   (b) among the triples in $zone(dist, q, S)$ with label l select the triple a that has the maximum weight ("witness count"); $S.add(a)$
   (c) `if` $S.size == k$ or G is exhausted `then return` S, `else` try to do next iteration of the while loop in line 2
3. $dist$++; $reset\ \Lambda$; `go to` line 2

Fig. 2. A greedy algorithm for DIVERSUM. $zone(i, q, S)$ (where $i \in N_+$) denotes the set of triples (x, a, y) in distance i from q and there is a path (that ignores arc directions) from x or y to q completely contained in S (i.e. a is connected to the result). First, the algorithm considers only candidates from $zone(1, q, S)$ (i.e. the triples adjacent directly to q) until the labels are exhausted, next, it focuses on $zone(2, q, S)$, $zone(3, q, S)$, etc. until the result consists of k triples or the underlying graph is exhausted. Λ - stands for the growing set of labels present in S. Notice that in line 3 we *reset* the Λ set. Different implementations of this operation allow for considering different variants of the algorithm. For example, if *reset* does nothing, we force uniqueness of each arc label selected for the summary; if *reset* makes the set empty, the uniqueness is forced only within each $zone$.

An example output of the algorithm is on figure 3 (right). One can see that the result does not suffer from the redundancy problem inherent to the PRECIS algorithm.

To summarise, from a high-level perspective, while PRECIS is totally diversity-unaware, the notion of diversity in the DIVERSUM algorithm proposed in [7] is extreme in the sense that it allows only *one repetition* of each relation type (arc label) within a fixed arc-node distance from the summarised entity.

Since the algorithms can be viewed as representing two opposite poles (in the above sense) in the context of entity summarisation on RDF graphs, the remaining sections (and the main focus of this paper) is devoted to perform and analyse user evaluation experiment to support our intuition that *diversification is the right direction towards improving the summarisation quality*.

4 Evaluation Experiment

The experiment had the following goals:

- Assessing the need for diversification in entity summarisation on RDF-graphs by comparison of diversity-aware algorithm (DIVERSUM), with a diversity-unaware baseline (PRECIS)
- Assessing absolute quality of the diversity-aware algorithm

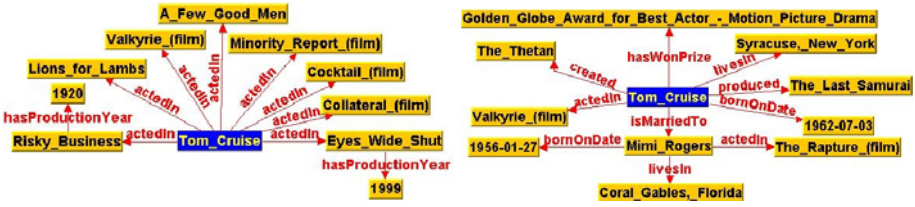


Fig. 3. Left: an example output of the PRECIS algorithm (noticeable redundancy in relation type). Right: an example output of the DIVERSUM algorithm. Both examples were computed for entity "Tom Cruise" and run on the IMDB dataset concerning movie domain with edge limit set to 10.

– Collecting valuable observations and user feedback, concerning both algorithms, that can serve for improving the summarisation algorithm in future work.

In the experiment, the PRECIS algorithm is used as a baseline graph entity summarisation algorithm to compare with the DIVERSUM algorithm. In the time of writing, the authors are not aware of any other publicly available algorithms that could be reasonably used as such a baseline.

A dataset for our experiments was extracted from the IMDB database concerning movie domain (`www.imdb.org`), that contains information concerning 59k entities, including 12k actors, over 530k edges representing 73 different binary relations. The "importance" weights on the arcs in this dataset were computed as inverse of, so called, "witness counts", (the number of times the given fact was found in the database).

To make the results visually cleaner, we pruned from the base graph some relations of technical nature, constituting redundant information or of little potential interest to the end users, so their omission would not affect the results significantly. The ignored IMDB relations are: *type*, *hasImdb*, *hasLanguage*, *hasProductionLanguage*, *hasDuration*, *participatedIn*. This pre-filtering step was done after manual inspection of many results of both algorithms. In addition, some triples were pruned since they represented some incorrect facts.[3]

Concerning the choice of entities to be summarised in the experiment, we selected 20 prominent actors contained in the dataset, with high overall number of movies the actor participated in and the condition that both considered algorithms can produce at least 14 edges (facts) in the summarisation.

The dataset contains also many other interesting types of entities to be potentially summarised, such as directors, for example. However, in order to obtain reliable results with limited resources, in this experiment we decided to focus only on a single type of entities being summarised: actors. Since the IMDB dataset contains thousands of actors, we first pre-selected about 50 most active actors and then manually selected 20 out of it, considering the number of edges produceable in their summaries. The set was intentionally diversified in the terms of geographical and cultural context. More precisely, it included not only actors that are known to the global audience but also some very active actors known only in some particular regions of the world (e.g. some prominent Indian

[3] Due to the imperfect knowledge harvesting procedure, such as entity disambiguation.

or European actors). This choice made it possible to better evaluate the summarisation algorithms in the full range of familiarity of the user with the summarised entity.

We decided to evaluate the algorithms for two different levels of summary edge limit: *low* (7 edges) and *high* (12 edges). The choice of number 12 was made after manual inspection of many summaries, as it turned out that summaries with substantially more than 12 edges are not easy to comprehend by humans (which is contradictory to the main function of a useful *summary*) and there was only a limited number of actors in the dataset for which both algorithms could generate much more than 12 edges.

The idea of the experimental evaluation was simple: to repeatedly present human evaluators with a pair of outputs produced by PRECIS and DIVERSUM, next to each other, concerning *the same entity* and the same number of presented facts, and ask simple questions concerning the relative and absolute quality of the outputs, such as:

- which output is preferred, why and what is the familiarity with the entity?
- what fraction of the presented facts is interesting or irrelevant to the user?
- what important facts about the entity are missing?
- what is the overall usefulness of the output as a small summary of the entity?

Before the experiment, the evaluators were given precise instructions, including a general description of the problem. To avoid any bias, the evaluators were not told any details concerning the summarising algorithms, and additionally, the outputs were presented each time *in random order*. Importantly, the evaluators were *not informed* that diversification is the issue studied in the experiment. In addition, the evaluators were noticed that the focus of the experiment is on the *choice of facts* in the summary rather than the particular graphical layout of the resulting graph.

Technically, the experiment was performed as follows. We designed and implemented a web application for presenting the pairs of outputs of the algorithms (in random order) and providing a form for collecting the user answers and open-text explanations for their the choice (figure 4).

Next, we invited a pool of evaluators that were not involved in this project and did not know its real goal. It is worth mentioning that the evaluators represented extreme

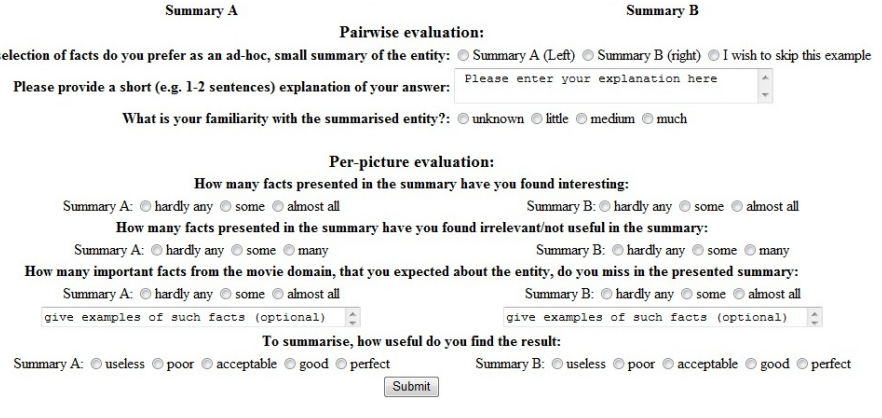


Fig. 4. Evaluation form used in the experiment. Due to space limitations, the picture omits summarisation outputs that were presented to the evaluator (see fig. 3 for an example of such pair.)

diversity in terms of cultural background and could be regarded as experts in the domain of information retrieval. Evaluators were our departmental colleagues, so that their objectivity and honesty can be assumed. In the experiment, the evaluators asynchronously visited the web page to assess the generated outputs. The evaluation was anonymysed, though from subsequent analysis of apache logs it seems that the number of actual active evaluators was between 10 and 20. The application was designed so that it was very unlikely that the same user would see the same pair of outputs more than once (in such case, the evaluator was instructed to skip the example).

5 Results and Discussion

Within three days of experiment's duration we collected 71 assessments out of which 66 expressed clear preference of one of the algorithms over another (5 other assessments were not completed). Each out of 20 selected actors received at least 2 assessments (18 actors received more) with median 3, mean 3.3, and maximum of 6.

As the main result of the experiment, in over 80% of cases, the diversity-aware summarisation (DIVERSUM) was preferred to the diversity-unaware one (figure 5). Importantly, the pairwise comparison results were consistent with the per-summary answers, i.e., whenever DIVERSUM was preferred, the corresponding answers for the absolute assessment of each algorithms also assessed DIVERSUM not lower than PRECIS.

Considering the dependence of the assessment on the summary size, DIVERSUM was even stronger preferred (over 90% of cases) for the low edge limit value ($k = 7$).

One of our doubts before the experiment was that the user assessments depend on the particular algorithms applied, not the "diversity" notion itself. Our fears were dispelled, though, because the evaluators independently happened to frequently notice and literally appreciate the higher degree of "diversity" when explaining why they prefer one algorithm (actually, DIVERSUM) over another. Remind that evaluators where *not* informed that diversification is the actual focus of the experiment.

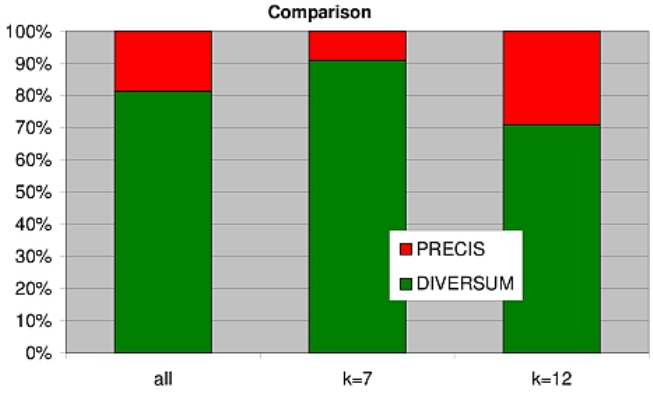


Fig. 5. Fraction of cases where human evaluators preferred DIVERSUM to PRECIS

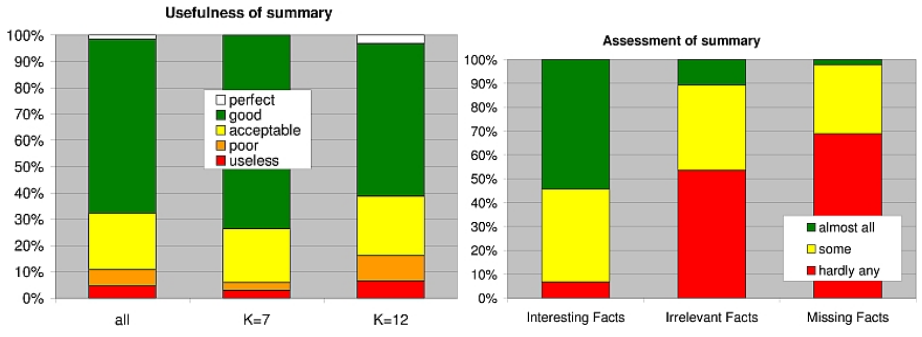


Fig. 6. Left: usefulness of the results of the DIVERSUM algorithm; right: assessment of facts selected by the DIVERSUM algorithm

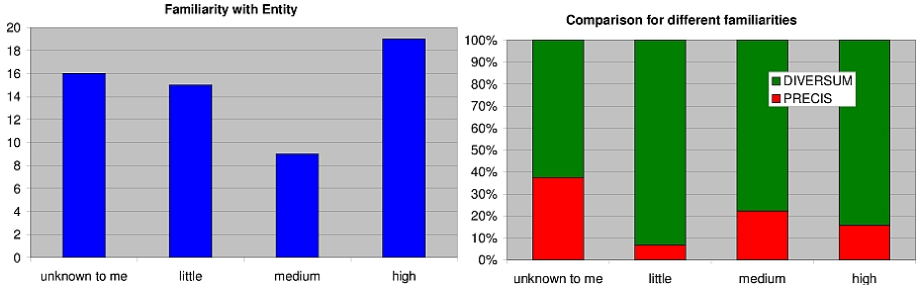


Fig. 7. Left: familiarity of the evaluators with the summarised entities (number of cases); right: preference between DIVERSUM and PRECIS for different levels of familiarity

Thus, the results of the experiment seem to really *demonstrate the value of diversification*, despite the obvious dependence of the outputs on particular algorithms used.

Considering the absolute assessments of the usefulness of the diversity-aware summarisation, in only 5% of cases it was marked as "poor" or "useless" for the low level of limit budget ($k = 7$) and in about 10% in total (figure 6). The average value for DIVERSUM was close to "good" (however there is still much room for improvement).

The above positive results of DIVERSUM are obtained for entities that are highly diversified in the terms of familiarity of the user with the summarised entity. More precisely, in about 50% of the cases the familiarity was marked as "high" or "medium" and in other as "little" or even "unknown" (see the left part of figure 7).

It is important to notice that DIVERSUM performs quite well, in opinion of the users, across the full range of degree of familiarity of the user with the summarised entity (see the right part of figure 7). In particular, for high familiarity it was preferred in about 83% cases, for little in over 90% cases. The algorithm seems to perform a bit worse for unknown entities, but it is still preferred here in about 62% of cases.

Considering more direct assessments of the facts selected by the DIVERSUM algorithm, the results are also very promising: in over 93% of cases the summary was assessed as having "almost all" or "some" interesting facts, and in about 98% of cases

Fig. 8. DIVERSUM: Comparison of assessment of selected facts for different levels on edge limit: $k = 12$ (left) and $k = 7$ (right). The layout of the graphs is the same as that of figure 6.

the evaluators did not complain that the summary was missing most important facts. In only about 11% of cases the summaries contained facts assessed as "irrelevant" by evaluators. See figure 6 for details.

There is a significant difference of user experience concerning the choice of facts selected by the DIVERSUM algorithm for two different levels of the edge limit (figure 8). Similarly to the previously reported aspects of summaries, the algorithm performs better for low level of edge limit (except the assessment of the "missing facts" as obviously performing better for higher edge limit).

Overall, one observes that the diversified entity summarisation algorithm received definitely better assessments than the diversity-unaware baseline in all evaluated aspects. It is clear that across all measured properties the attractiveness of the diversified summarisation is higher for smaller summarisation size (limit). It has an intuitive explanation: as the number of possible facts to select decreases, the algorithm should pay more attention to try to cover more diversified aspects of the entity.

To summarise, the results of the experiment constitute an evidence that:

- diversification of the results of entity summarisation is a very appreciated property
- the diversity-aware summarisation algorithm (DIVERSUM) is significantly prefered by human evaluators over the diversity-unaware one
- the DIVERSUM algorithm obtained quite high absolute notes from user evaluators.
- the diversity of summarisation is even more appreciated for low limit of facts that can be presented in the summary.

We observed the following problems while analysing the results:

- some results seem to suffer a "topic drift" i.e. presenting some facts that seem to be too distant from the entity summarised
- sometimes, the algorithm selects irrelevant facts

It seems that both problems are, to a large extent, caused by the fact that the current variant of DIVERSUM does not allow for repeated selection of the same arc label in the given "zone". This property can be viewed as "extreme diversification" and actually seems to be too strong. As a consequence, when each arc label in the current zone is represented by some selected fact, the algorithm starts to look for facts that are in further zones, that would result in topic drift or presenting some irrelevant facts.

This problems can be addressed in an improved variant of the algorithm, by allowing for a mild arc label repetition in a given zone. In addition, such a "relaxation" of the algorithm may produce better results, since it would be possible to better represent the characteristic of a given entity by showing more facts that concern the main activity of this entity (e.g. more facts concerning "acted in" for a famous, active actor).

Finally, only once the usefulness of a summary was marked as "perfect" which means that there is still room for improvement.

6 Conclusions and Further Work

We studied the issue of diversified entity summarisation in knowledge graphs and presented experimental user evaluation. Our experiments performed on real IMDB dataset, clearly show that diversity is a needed property of a summarisation and, in addition, the DIVERSUM algorithm obtained quite positive absolute feedback that encourages for continuing the work. One of the main future directions of future work, due to our observations, will be to allow for a mild arc label repetition to alleviate the problems of topic drift and irrelevant fact selection. It would be valuable to experiment with other types of entities (e.g., directors or movies) and on datasets from different domains (e.g., the libraryThing dataset from [4]). Importantly, in the current approach there is no objective measure of RDF-graph summary diversity introduced. Designing such a measure is a part of ongoing work [9]. Another direction is to improve the automated visualisation of the result. In such an automated visualisation, the arc labels could be automatically grouped (e.g. facts concerning private life of an actor vs his professional life) based on some automatically collected statistics such as arc label co-incidence. It would be also very interesting to directly incorporate user interest profiles to a future version of our algorithm towards personalizing the resulting summarisation.

Acknowledgements. The first author was supported by the DAAD Prize for visiting MPII, Saarbruecken in 2010, when part of this work was done and the N N516 481940 grant of the Polish Ministry of Science and Higher Education. The authors would like to thank all the volunteers from MPII that participated in the experimental evaluation.

References

1. Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying search results. In: WSDM, pp. 5–14 (2009)
2. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–336. ACM, New York (1998)
3. Clarke, C.L.A., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttche, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: SIGIR, pp. 659–666 (2008)
4. Elbassuoni, S., Ramanath, M., Schenkel, R., Sydow, M., Weikum, G.: Language-model-based ranking for queries on rdf-graphs. In: CIKM, pp. 977–986 (2009)

5. Ramanath, M., Kumar, K.S.: Xoom: a tool for zooming in and out of xml documents. In: EDBT, pp. 1112–1115 (2009)
6. Ramanath, M., Kumar, K.S., Ifrim, G.: Generating concise and readable summaries of xml documents. In: CoRR, abs/0910.2405 (2009)
7. Sydow, M., Pikuła, M., Schenkel, R.: DIVERSUM: Towards diversified summarisation of entities in knowledge graphs. In: Proceedings of Data Engineering Workshops (ICDEW) at IEEE 26th ICDE Conference pp. 221–226. IEEE, Los Alamitos (2010)
8. Sydow, M., Pikuła, M., Schenkel, R., Siemion, A.: Entity summarisation with limited edge budget on knowledge graphs. In: Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 513–516. IEEE, Los Alamitos (2010)
9. Sydow, M.: Towards the Foundations of Diversity-Aware Node Summarisation on Knowledge Graphs. In: The Proceedings of the "Diversity in Document Retrieval" Workshop at the ECIR 2011 Conference (to appear, 2011)
10. Vee, E., Srivastava, U., Shanmugasundaram, J., Bhat, P., Amer-Yahia, S.: Efficient computation of diverse query results. In: ICDE, pp. 228–236 (2008)
11. Wan, X.: Topic analysis for topic-focused multi-document summarization. In: CIKM, pp. 1609–1612 (2009)
12. Wan, X., Xiao, J.: Exploiting neighborhood knowledge for single document summarization and keyphrase extraction. ACM Trans. Inf. Syst. 28(2) (2010)
13. Zhang, N., Tian, Y., Patel, J.M.: Discovery-driven graph summarization. In: ICDE, pp. 880–891 (2010)

Improving the Accuracy of Similarity Measures by Using Link Information

Tijn Witsenburg[1] and Hendrik Blockeel[1,2]

[1] Leiden Institute of Advanced Computer Science, Universiteit Leiden
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
tijn@liacs.nl, blockeel@liacs.nl
[2] Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, 3001 Leuven, Belgium
hendrik.blockeel@cs.kuleuven.be

Abstract. The notion of similarity is crucial to a number of tasks and methods in machine learning and data mining, including clustering and nearest neighbor classification. In many contexts, there is on the one hand a natural (but not necessarily optimal) similarity measure defined on the objects to be clustered or classified, but there is also information about which objects are linked together. This raises the question to what extent the information contained in the links can be used to obtain a more relevant similarity measure. Earlier research has already shown empirically that more accurate results can be obtained by including such link information, but it was not analyzed why this is the case. In this paper we provide such an analysis. We relate the extent to which improved results can be obtained to the notions of homophily in the network, transitivity of similarity, and content variability of objects. We explore this relationship using some randomly generated datasets, in which we vary the amount of homophily and content variability. The results show that within a fairly wide range of values for these parameters, the inclusion of link information in the similarity measure indeed yields improved results, as compared to computing the similarity of objects directly from their content.

1 Introduction

Similarity is an important notion in machine learning and data mining; it plays a crucial role in, for instance, many clustering methods, and in nearest neighbor classification. The quality of obtained clusterings or classifiers depends strongly on whether a relevant similarity measure is used ("relevant" meaning suitable for the task at hand).

In application domains such as web mining or social network analysis, objects do not only have an internal content, they are also linked together by a graph structure. The question is then, whether this graph structure can help clustering or classification. This has already been shown convincingly in several research areas, such as collective classification [4] or entity resolution [5]. In this paper,

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 501–512, 2011.

we investigate, in a more general context, one way in which a better similarity measure can be devised by incorporating network information.

To make this more concrete, consider two web pages, both about the same subject, but far away from each other in the web. These are highly unlikely to be clustered together by standard graph clustering systems (because graph clustering methods typically form clusters of nodes that are close to each other in the graph), yet we want to consider them as similar. Their contents will indicate that they are similar indeed. One could argue that looking only at the content, and ignoring the graph structure, is then likely to be the most appropriate way of comparing the pages. In this paper, we show that this is not the case: the use of some information in the graph structure, by other means than a standard graph clustering algorithm, can yield a more relevant similarity measure than what we can compute based solely on the object's content. We explain why this may be the case, and investigate the conditions under which it is indeed the case. This investigation includes both a theoretical analysis and an experiment with synthetic data that allows us to quantify the predicted effects.

We will discuss the problem in the context of annotated graphs, an abstract formalism that generalizes over more concrete application domains such as social networks, protein protein interaction networks, digital libraries, or the world wide web. We introduce the problem setting formally in Section 2; next, in Section 3, we will discuss how similarity measures can be improved by taking the network structure into account. Section 4 reports on an experiment with synthetic data that quantifies the effects discussed in Section 3. Section 5 briefly discusses related work, and Section 6 concludes the paper.

2 Problem Setting

2.1 Annotated Graphs

We formulate the problem formally using the concept of an annotated graph. An annotated graph is a graph where nodes or edges can have annotations, which can be of a varying nature, and more complex than simple labels. An example of an annotated graph is the web, where edges are hyperlinks and nodes are html documents. Other examples include social networks (where edges are friend relationships and nodes are people; a person can have a rich description) and digital libraries (edges are references from one paper to another). In this paper, we will consider only node annotations, not edge annotations; this covers, among other, all the above applications.

Formally, we define an annotated graph G as a simple undirected graph $G = (V, E, \lambda)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of n vertices or data elements, $E \subseteq V \times V$ is the set of edges, and $\lambda : V \to \mathcal{A}$ is a function that assigns to any element v of V an "annotation". We also call this annotation $\lambda(v)$ the *content* of vertex v. As the graph is simple (no loops) and undirected, we have for all $v, w \in V$ that $(v, w) \in E \Leftrightarrow (w, v) \in E$ and $(v, v) \notin E$.

The space of possible annotations is left open; it can be a set of symbols from, or strings over, a finite alphabet; the set of reals; an n-dimensional Euclidean

space; a powerset of one of the sets just mentioned; etc. The only constraint on $\mathcal{A}$ is that it must be possible to define a similarity measure $s_{\mathcal{A}} : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ that assigns a value to any pair of annotations expressing the similarity between these annotations.

2.2 Semantic and Observed Similarity

All data elements are described using an annotation, which is an element of $\mathcal{A}$. In general, it is useful to distinguish the representation of an object from the object itself. For instance, in a dataset of objects that are to be clustered, each object might be described using a fixed set of attributes, but that does not necessarily imply that these attributes completely determine the object; rather, an object's representation is a projection of the object onto the representation space. Similarly, when we have a text document, we can associate a certain semantics to the document; when representing the document as a bag of words, it is projected onto a vector space, and different documents (with different semantics) are projected onto the same vector. (E.g., "Carl hit the car" means something different from "The car hit Carl", but the bag-of-words representation of these two sentences is exactly the same.)

This discussion shows that similarity in the annotation space does not necessarily coincide with semantic similarity: it is only an approximation of it. In practice, when we perform clustering or similarity-based classification (as in nearest neighbor methods), our real intention is to get results that are semantically meaningful.

We call the semantic space (in which the real objects live) $\mathcal{S}$; an object $o \in \mathcal{S}$ is represented using an annotation $a \in \mathcal{A}$. Besides the similarity measure $s_{\mathcal{A}}$ in $\mathcal{A}$, we assume there is a similarity measure in $\mathcal{S}$, $s_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, which expresses the semantic similarity of two objects.

Note that, in general, it will be difficult to define $\mathcal{S}$ and $s_{\mathcal{S}}$ in a formal way. We assume that they exist in the real world, that there is some intuitive notion of them; if not, we would not be able to talk about the "meaning" of the sentence "the car hit Carl", and how it differs from the meaning of "Carl hit the car". But they may be very difficult to specify: for instance, given a picture, what is the semantics of it, i.e., what does it represent? (One approximation might be constructed, for instance, by asking multiple people what they see and experience when looking at the picture and take the intersection of all their comments.) Generally, the semantics of an object is very difficult to define, but for this paper, we do not need to do that; we just assume that there is some notion of semantics and semantic similarity, and it is expressed abstractly by $\mathcal{S}$ and $s_{\mathcal{S}}$.

2.3 Similarity Approximations

The fact that there is a difference between observed similarity $s_{\mathcal{A}}$ and semantic similarity $s_{\mathcal{S}}$, and the fact that we are really interested in semantic similarity, raises the question: can we find a better approximation to the semantic similarity than $s_{\mathcal{A}}$? This would of course be possible by changing the annotation space $\mathcal{A}$,

making it richer; but we assume that $\mathcal{A}$ and the data elements in it are given and cannot be changed. (The question of how to change $\mathcal{A}$ is essentially the feature construction problem, which is orthogonal to what we discuss in this paper.) However, when the documents are also linked together, we may be able to obtain a better approximation of $s_{\mathcal{S}}$ by taking the link structure into account. In the following section, we propose a simple way of doing that.

3 Similarity Measures

3.1 Content-Based, Contextual, and Combined similarity

We now discuss three similarity measures for annotated nodes in a graph. These similarity measures were first proposed in [6].

The first is called **content similarity**; it is identical to the $s_{\mathcal{A}}$ similarity measure proposed before:

$$S_{content}(v, w) = s_{\mathcal{A}}(\lambda(v), \lambda(w))$$

Now let $\phi : V \times V \to \{0, 1\}$ be a function that assigns a value to a pair of elements in the data set such that $\phi(v, w) = 1$ if $(v, w) \in E$ and 0 otherwise. We define the *neighbor similarity* $S_{neighbor} : V \times V \to \mathbb{R}$ between two elements v and w from V as the average annotation similarity between v and all neighbors of w:

$$S_{neighbor}(v, w) = \frac{\sum_{u \in V} s_{\mathcal{A}}(\lambda(v), \lambda(u)) \cdot \phi(u, w)}{\sum_{u \in V} \phi(u, w)} \tag{1}$$

This similarity is not symmetric, but we can easily symmetrize it, leading to the **contextual similarity** $S_{context} : V \times V \to \mathbb{R}$:

$$S_{context}(v, w) = \frac{S_{neighbor}(v, w) + S_{neighbor}(w, v)}{2} \tag{2}$$

The motivation behind defining this similarity measure is, briefly: if similar nodes tend to be linked together, then the neighbors of w in general are similar to w; and if similarity is transitive, a high similarity between v and many neighbors of w increases the reasons to believe that v is similar to w, even if there is little evidence of such similarity when comparing v and w directly (for instance, due to noise or missing information in the annotation of w). In the following section, we explain this motivation in more detail.

This contextual similarity measure is complementary to the content-based one, in the sense that it does not use the content-based similarity between v and w at all. Since in practical settings it may be good not to ignore this similarity entirely, it may be useful to consider the **combined similarity** $S_{combined} : V \times V \to \mathbb{R}$:

$$S_{combined}(v, w) = c \cdot S_{content}(v, w) + (1 - c) \cdot S_{context}(v, w) \tag{3}$$

with $0 \leq c \leq 1$. c determines the weight of the content-based similarity in the combined similarity. As Witsenburg and Blockeel [6] found no strong effect of

using different values for c, from now on we consider only the combined similarity with $c = \frac{1}{2}$.

We call the contextual and the combined similarity measures *hybrid* similarity measures, because they use information in the graph structure as well as in the annotations.

3.2 Intuitive Motivation for Hybrid Similarity

Intuitively, the motivation for the use of hybrid similarity measures relies on three phenomena that can be expected to occur in an annotated graph (such as the web or a social network):

- **Homophily in the network:** homophily refers to the fact that in a network, connections are more likely between similar objects than between dissimilar objects; for instance, people who are more similar (same age, common interests, etc.) are more likely to be friends. Homophily is a phenomenon that often occurs in social and other networks [2].
- **Transitivity of similarity:** with this we mean that if x and y are similar, and y and z are similar, then x and z will be relatively similar as well. Note that transitivity is in some sense a generalization of the triangle inequality for distances. Indeed, when similarity is expressed using a distance metric (shorter distance implying higher similarity), for instance, $s(x, y) = 1/d(x, y)$ with d some distance metric, then $d(x, z) \leq d(x, y) + d(y, z)$ immediately leads to $s(x, z) \geq 1/(1/s(x, y) + 1/s(y, z))$, that is, $s(x, z)$ is lower bounded by half the harmonic mean of $s(x, y)$ and $s(y, z)$, which gives a concrete quantification of transitivity in this particular case. Similarity measures do not have to be transitive in any predefined way, but some intuitive notion of transitivity is generally included in the intuition of similarity.
- **Content variability:** when two nodes represent the same semantic object o, the actual content of the nodes (their annotation) may still vary. Generally, it can be assumed that the content of a node varies around some center; this central representation could be seen as the "closest" approximation to the semantic object. The variation in possible content of a node, given the semantic object it refers to, creates an inherent lower bound on the expected error of any estimator that estimates the object from the annotation alone.

The combination of these three phenomena makes it possible to explain why a contextual similarity might work better than a pure content-based similarity. Essentially, the estimated similarity between two objects, as measured by the content-based similarity, will have an error because of content variability. Homophily and transitivity imply that when an object is similar to another object, it is likely to be similar also to that objects neighbors, or more generally: the similarity between o_1 and any neighbor of o_2 will not differ very much from the similarity between o_1 and o_2 itself. Now the point is that, since o_2 can have multiple neighbors, a statistical effect comes into play: the average similarity between o_1 and the neighbors of o_2 can be estimated more accurately than an individual similarity between o_1 and a neighbor of o_2, or than the similarity between o_1 and o_2. We describe this effect mathematically in the following section.

3.3 Analysis of Hybrid Similarity

Let us denote the semantic similarity between the contents of two nodes v and w as $\sigma(v,w)$; that is, $\sigma(v,w) = s_{\mathcal{S}}(\lambda(v),\lambda(w))$. Further, we use the notation $\hat{\sigma}$ to denote an estimator of σ. The content-based similarity $S_{content}$ is one such estimator; we denote it $\hat{\sigma}_1$. Let ϵ denote the difference between this estimator and what it estimates. We then have

$$\hat{\sigma}_1(v,w) = S_{content}(v,w) = \sigma(v,w) + \epsilon(v,w). \tag{4}$$

Any distribution over pairs of annotated nodes gives rise to distributions of σ, $\hat{\sigma}$ and ϵ. If the estimator is unbiased, ϵ has a distribution centered around 0: $E[\epsilon(v,w)] = 0$. Note that any estimator can be made unbiased by subtracting its bias from it; so in the following, we will simply assume that the estimator is unbiased, hence $E[\epsilon(v,w)] = 0$. Finally, we denote the variance of this distribution as σ_ϵ^2.

Now consider, as a second estimator, the neighbor similarity:

$$\hat{\sigma}_2(v,w) = S_{neighbor}(v,w) = \frac{\sum_i S_{neighbor}(v,w_i)}{n} = \frac{\sum_i \sigma(v,w_i)}{n} + \frac{\sum_i \epsilon(v,w_i)}{n} \tag{5}$$

where w_i ranges over all neighbors of w (i.e., $\sum_i$ is short for $\sum_{w_i|(w,w_i)\in E}$) and n is the number of such neighbors.

Now consider the conditional distribution of $\sigma(v,w_i)$, given $\sigma(v,w)$. $\sigma(v,w_i)$ is likely dependent on $\sigma(v,w)$, since w and w_i are connected. More specifically, due to homophily and transitivity, as explained before, we expect them to be positively correlated. Let us write

$$\sigma(v,w_i) = \sigma(v,w) + \xi(w,w_i) \tag{6}$$

where ξ denotes the difference between, on the one hand, the similarity between v and w, and on the other hand, the similarity beween v and w_i. This formula is generally valid (by definition of ξ), but in the presence of homophily and transitivity, we additionally expect ξ to be small.

Using Equation 6 to substitute $\sigma(v,w_i)$ in Equation 5 gives

$$\hat{\sigma}_2(v,w) = \sigma(v,w) + \frac{\sum_i \xi(w,w_i)}{n} + \frac{\sum_i \epsilon(v,w_i)}{n}. \tag{7}$$

Observe that, among v, w and w_i there is no relation except for the fact that w and w_i are connected to each other. This entails the following. For symmetry reasons, the distribution of $\epsilon(v,w_i)$ must be equal to that of $\epsilon(v,w)$ (from v's point of view, w_i is just a random node, just like w). For each w_i, $\epsilon(v,w_i)$ therefore has an expected value of 0 and a variance of σ_ϵ^2.

Again because of symmetry, there is no reason to believe that $\sigma(v,w)$ should on average be greater or smaller than $\sigma(v,w_i)$, which means $E[\xi(w,w_i)] = 0$. As the w_i are also interchangeable among themselves, all $\xi(w,w_i)$ have the same distribution, the variance of which we denote as σ_ξ^2.

Now consider the case where all $\epsilon(v, w_i)$ are independent, all $\xi(w, w_i)$ are independent, and $\epsilon(v, w_i)$ is independent from $\xi(w, w_i)$ for all w_i. For this special case, we obtain the following squared errors for the two estimators:

$$SE(\hat{\sigma}_1(v, w)) = E[(\hat{\sigma}_1(v, w) - \sigma(v, w))^2] = \sigma_\epsilon^2$$

$$SE(\hat{\sigma}_2(v, w)) = E[(\hat{\sigma}_2(v, w) - \sigma(v, w))^2] = \frac{\sigma_\epsilon^2 + \sigma_\xi^2}{n}$$

We now see that the first estimator has an expected squared error of σ_ϵ^2, whereas the second estimator has an expected squared error of $(\sigma_\epsilon^2 + \sigma_\xi^2)/n$. This second term can be larger or smaller than the first; it tends to become smaller as n, the number of neighbors, increases, but when (and whether) it becomes smaller than σ_ϵ^2 depends on the relative size of σ_ξ^2.

Note that, intuitively, σ_ϵ^2 is related to content variability (it reflects the extent to which the observed content similarity between two nodes approximates their real similarity), whereas σ_ξ^2 is related to the amount of homophily in the network (it expresses to what extent nodes that are linked together are indeed similar). In networks with strong homophily and high content variability, the second estimator can be expected to be more accurate than the first.

The case where ϵ and ξ are not independent is mathematically more complex. We already have σ_ϵ^2 and σ_ξ^2 for the variance of $\epsilon(v, w_i)$ and $\xi(w, w_i)$. If we denote the covariance between $\epsilon(v, w_i)$ and $\epsilon(v, w_j)$ $(j \neq i)$ as C_ϵ, the covariance between $\xi(w, w_i)$ and $\xi(w, w_j)$ $(j \neq i)$ as C_ξ, and the covariance between $\epsilon(v, w_j)$ and $\xi(w, w_i)$ as $C_{\epsilon\xi}$, then using the rule that $Var(\sum_i X_i) = \sum_i Var(X_i) + \sum_{i,j \neq i} Cov(X_i, X_j)$, and exploiting the linearity of Var and Cov, one quickly arrives at

$$SE(\hat{\sigma}_2(v, w)) = \frac{\sigma_\epsilon^2}{n} + \frac{\sigma_\xi^2}{n} + \frac{n-1}{n} C_\epsilon + \frac{n-1}{n} C_\xi + \frac{2}{n} C_{\epsilon\xi} \tag{8}$$

which shows that strong autocorrelations of ξ and ϵ are likely to spoil the advantage of using $\hat{\sigma}_2$. Such correlations are not impossible; for instance, when $\epsilon(w, w_1)$ is high, this may be because $\lambda(w)$ deviates strongly from its expected value (due to content variability), which makes $\epsilon(w, w_2)$ more likely to be high too. It is difficult to estimate how large this effect can be.

Finally, note that $S_{context}(v, w)$ is the average of $S_{neighbor}(v, w)$ and $S_{neighbor}(v, w)$, and hence is likely to be slightly more accurate than $\hat{\sigma}_2$ (its standard error is $\sqrt{2}$ times smaller, in case of independence of the error terms for v and w), again due to the error-reducing effect of averaging. $S_{combined}$, being the average of $S_{content}$ and $S_{context}$, can be more accurate than either or them, or have an accuracy somewhere in between.

4 Experiments with Synthetic Data

4.1 The Synthetic Data Set

In order to test the relative performance of $S_{content}$, $S_{context}$, and $S_{combined}$, similarity measures proposed above, and characterize under which circumstances

which measure works best, we have created a synthetic data set in which we can vary the amount of content variability and homophily in the network. Roughly, one could think of the data set as a set of documents; each document is represented as a boolean vector that shows which words occur in the document, and documents may be linked to each other. The documents are organized into classes. We start with a dataset with zero content variability and perfect homophily: each document of a particular class is the same (has all the words characteristic for the class, and no words characteristic for other classes), within a class all documents are linked to each other, and there are no links between classes. Afterwards, we increase content variability by removing words from documents and introducing words from other classes into them. Similarly, we decrease homophily by introducing links between documents of different classes and removing links between documents from the same class. Our goal is to investigate the effect of these changes on the accuracy of the similarity measures.

Concretely, our dataset D consists of 1000 data elements $\mathbf{x}_i$, $i = 1, \ldots, 1000$, divided into 10 classes of 100 elements each. Each element is a binary vector with 1000 components, each of which represents one particular word; each class has 100 words that are characteristic for it, and each word is characteristic for only one class. A 1000 by 1000 binary matrix G indicates the connections between the data elements. In the original dataset, D and G are both block matrices; for $100(c-1)+1 \leq i \leq 100c$, $x_{ij} = 1$ for $j = 100(c-1)+1, \ldots, 100c$ and 0 elsewhere (that is, for each class all documents contain all characteristic keywords and no other), and similar for G (all and only documents of the same class are connected).

Now, increasing content variability means flipping bits in D, and decreasing homophily means flipping bits in G. The amount of bits that is flipped is determined by a parameter p (where relevant we write p_D and p_G for the corresponding parameter for D and G), with $0 \leq p \leq 1$. Originally, we flipped bits randomly, giving each single bit a probability p of being flipped, but this led to artifacts in the results, due to the fact that each document has approximately the same amount of flipped bits (a narrow binomial distribution around 1000p). Therefore, instead, we have chosen p to be the maximum fraction of bits flipped, but to let the actual number of flipped bits vary between 0 and $\Psi = 1000p$ according to a sinus-shaped distribution:

$$f(x) = \begin{cases} \frac{\pi}{2\Psi} \sin\left(\frac{\pi x}{\Psi}\right) & \text{if } x \in [0, \Psi] \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

This distribution (visualized in Figure 1) was chosen ad hoc; the actual distribution does not matter much, the important thing is to have an easily computable distribution that is not too narrow and gradually approaches zero near the borders.

We will alter p_D and p_G separately. We expect that increasing p_D (content variability) renders the content based similarity less accurate, while the other similarity suffer less from this. Conversely, increasing p_G is expected to have no influence on content based similarity, but cause the other similarities to deteriorate.

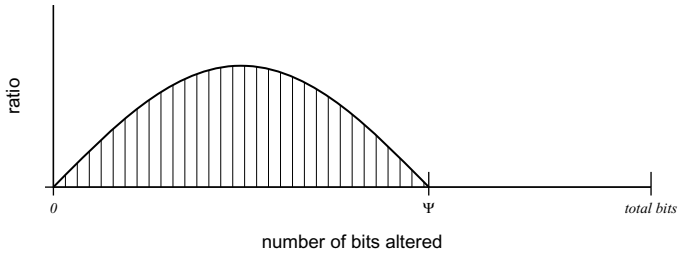


Fig. 1. Distribution of the amount of bits that will be flipped

4.2 Clustering Methods

We can evaluate the practical usefulness of the similarity measures by using them to perform clustering; a similarity measure is considered more relevant if it yields clusters that are closer to the actual (hidden) classes. We have used three different clustering algorithms: agglomerative hierarchical clustering using complete linkage (hereafter referred to as agglomerative clustering), k-means and k-medoids.

We express the quality of a clustering using the *F-measure*, as suggested by Larsen and Aone [1]. The F-measure for a cluster c with respect to a class l is defined as $F(c,l) = \frac{2PR}{P+R}$, where P is the precision (number of documents labelled l in a cluster c, divided by the number of documents in c) and R is the recall (number of documents labelled l in cluster c, divided by the total number of documents of class l in the data set). Larsen and Aone define the F-measure of a class as the highest F-measure found for that class for any cluster, i.e., $F(c) = \max_l F(c,l)$, and the *overall F-measure* of the clustering as the weighted average of all $F(c)$ (weighted according to the number of instances belonging to that class). Its value is between 0 and 1, where 1 is a perfect score.

Since there are 10 classes in the data set, we choose $k = 10$ for k-means and k-medoids, while the hierarchical clustering is simply cut off at 10 clusters.

4.3 Results

We ran experiments for many combinations of p_D and p_G. Figure 2 shows several samples from the obtained results. We first discuss those that show the effect of content variability p_D, while the homophily parameter p_G remains constant; these are shown in the left column in Figure 2 for, from top to bottom: agglomerative clustering with $p_G = 30\%$, k-means with $p_G = 60\%$, and k-medoids with $p_G = 0\%$. The curves are as expected: they start near 1 at the left; as p_D increases, all measures eventually yield less good results; however, some go down faster than others. This is most visible for k-means: here the purely contextual measure keeps giving good performance when the other deteriorate. For k-medoids and agglomerative clustering, the effect is much less outspoken, yet, contextual similarity tends to be equally good as, or better than, the other measures.

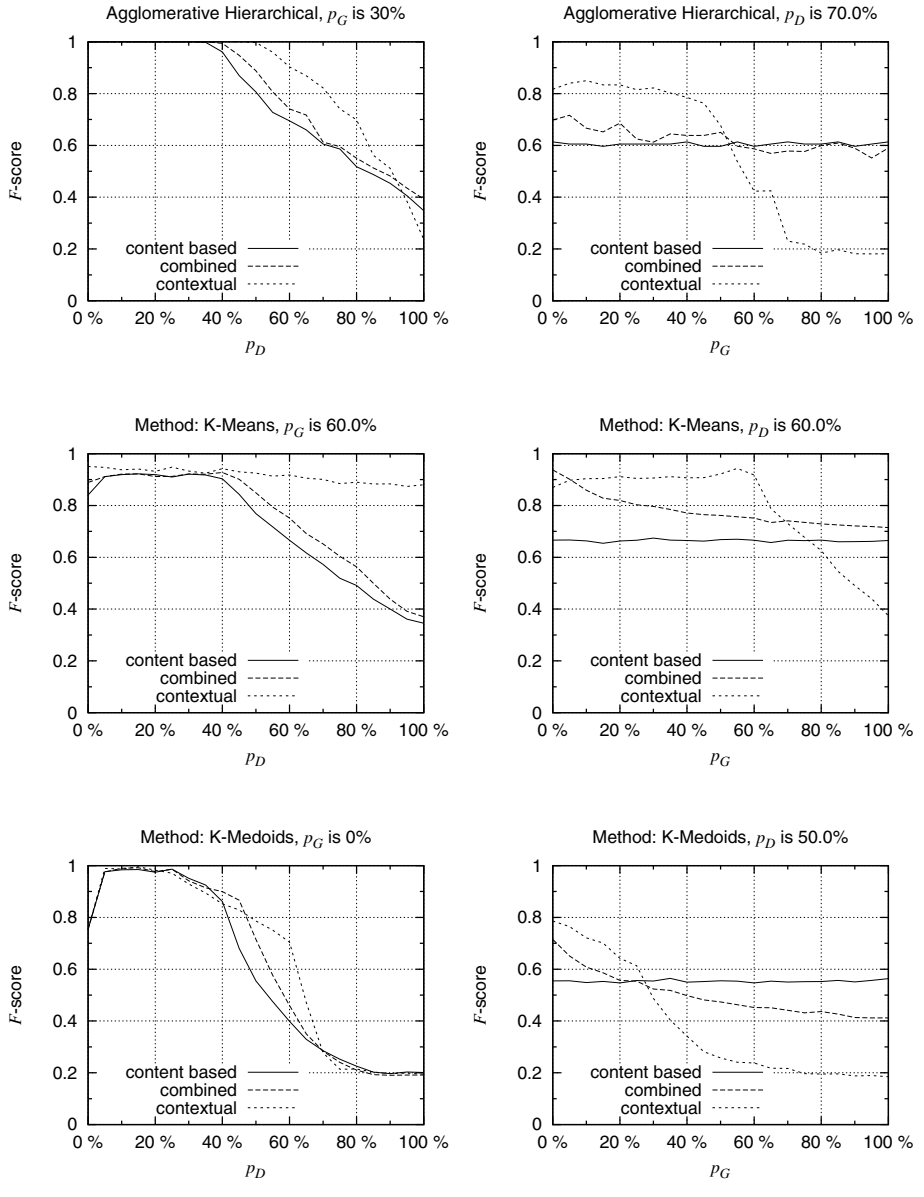


Fig. 2. Each graph plots for each of the three similarity measures the overall F-score against a varying p_D (left) or p_G (right) parameter, and this for agglomerative clustering (top row), k-means (middle row) and k-medoids (bottom row)

We now turn to the effect of p_G. Here we expect the F-measure of the found clusterings to decrease with increasing p_G for the contextual similarity, less so for the combined similarity, and to remain constant for content based similarity. The right column in Figure 2 shows results for (top to bottom) agglomerative clustering with $p_D = 70\%$, k-means with $p_D = 60\%$, and k-medoids with $p_D = 50\%$. In all graphs, the contextual similarity clearly outperforms the content based similarity when $p_G = 0\%$, but the opposite is true when $p_G = 100\%$. The point where they cross is fairly early for k-medoids, about halfway for agglomerative clustering, and fairly late for k-means. This was consistently the case also for other values of p_D (not shown).

4.4 Results on Real Data

The above analysis used synthetic data, where homophily and content variability were controlled. It is clear that, depending on the amount of homophily and content variability, one method will be better than the other. The question remains in which areas certain real-life datasets lie. Earlier work with the Cora dataset [6] has shown that the area where contextual and combined similarity perform better is certainly not empty. This, in turn, implies that some networks indeed contain enough homophily to make the use of hybrid similarities rewarding.

5 Related Work

We have no knowledge of other work that discusses, given the difference between observed and semantic similarity of objects, how a better estimate of the semantic similarity can be obtained by using information in the network. The most closely related work is that on clustering annotated nodes in a graph. Witsenburg and Blockeel [6,7] distinguish clustering methods that look only at node content, clustering methods that look only at the graph structure, and clustering methods that use both. Among the latter, those by Neville et al. [3] and Zhou et al. [8] are most relevant, as they can be seen as hybrid clustering methods. They address the clustering problem itself (reducing it to a standard graph clustering problem); they do not address the more basic question of how to improve similarity estimates for objects using their graph context.

6 Conclusion

When estimating the similarity between two annotated nodes in a graph, one can simply rely on the nodes' annotations. However, we have argued in this paper that it may be useful to take also the network context into account. For one particular kind of similarity, we have argued, both on an intuitive level and using a more detailed analysis, under what conditions this may be the case; we have related this to the properties of homophily, content variability, and similarity of transitivity. Using a synthetic dataset in which homophily and content variability are controlled, we have observed experimentally that the similarity measures

indeed behave as expected by our theoretical analysis. This confirms that, and explains why, similarity measures that look not only at content but also at graph context are of practical importance.

Acknowledgements. This research is funded by the Dutch Science Foundation (NWO) through a VIDI grant.

References

1. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 16–22 (1999)
2. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual Review of Sociology 27, 414–444 (2001)
3. Neville, J., Adler, M., Jensen, D.: Clustering relational data using attribute and link information. In: Proceedings of the Text Mining and Link Analysis Workshop, Eighteenth International Joint Conference on Artificial Intelligence (2003)
4. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classication in network data. AI Magazine 29(3), 93–106 (2008)
5. Singla, P., Domingos, P.: Entity resolution with markov logic. In: Proceedings of the Sixth International Conference on Data Mining, ICDM 2006, pp. 572–582. IEEE Computer Society Press, Los Alamitos (2006)
6. Witsenburg, T., Blockeel, H.: A method to extend existing document clustering procedures in order to include relational information. In: Kaski, S., Vishwanathan, S., Wrobel, S. (eds.) Proceedings of the Sixth International Workshop on Mining and Learning with Graphs, Helsinki, Finland (2008)
7. Witsenburg, T., Blockeel, H.: K-means based approaches to clustering nodes in annotated graphs (2011)(submitted)
8. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. In: Proceedings of the VLDB Endowment, Lyon, France, pp. 718–729. VLDB Endowment (2009)

Repetition and Rhythmicity Based Assessment Model for Chat Conversations

Costin-Gabriel Chiru[1], Valentin Cojocaru[1], Stefan Trausan-Matu[1,2], Traian Rebedea[1], and Dan Mihaila[1]

[1] "Politehnica" University of Bucharest, Department of Computer Science and Engineering, 313 Splaiul Independetei, Bucharest, Romania
[2] Research Institute for Artificial Intelligence of the Romanian Academy, 13 Calea 13 Septembrie, Bucharest, Romania
costin.chiru@cs.pub.ro, valentin.cojocaru@cti.pub.ro, {stefan.trausan,traian.rebedea,dan.mihaila}@cs.pub.ro

Abstract. This paper presents a model and an application that can be used to assess chat conversations according to their content, which is related to a number of imposed topics, and to the personal involvement of the participants. The main theoretical ideas that stand behind this application are Bakhtin's polyphony theory and Tannen's ideas related to the use of repetitions. The results of the application are validated against the gold standard provided by two teachers from the Human-Computer Interaction evaluating the same chats and after that the verification is done using another teacher from the same domain. During the verification we also show that the model used for chat evaluation is dependent on the number of participants to that chat.

Keywords: Rhythmicity, Polyphony, Chat Analysis, Repetition, Involvement, Computer-Supported Collaborative Learning, Dialogism.

1 Introduction

Lately, one can see a tendency toward an increased use of collaborative technologies for both leisure and work. There is an intense use of instant messaging systems (chats), blogs, forums, etc. for informal talks in our spare time. Their usage is also encouraged by the learning paradigm of Computer-Supported Collaborative Learning (CSCL) that suggests these tools are also suitable for collaborative knowledge building: "many people prefer to view learning as becoming a participant in a certain discourse" [10, 13].

Unfortunately, these tools do not provide analysis facilities that keep up with the above mentioned tendency, and therefore nowadays there are a lot of collaborative conversations that cannot be assessed – one cannot say whether one such conversation was good/efficient or not and is also unable to evaluate the participation of every participant.

Most of the research done in conversations' analysis is limited to a model with two interlocutors where at all moments there is usually only one topic in focus. The

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 513–522, 2011.

analysis is often based on speech acts, dialog acts or adjacency pairs [6]. Most of the time, the analysis is done to detect the topics discussed and to segment the conversation [1, 9] or to identify the dialogue acts [7].

However, there are situations when more than two participants are involved in a conversation. This claim is obvious for forums, but is also valid for chats allowing explicit referencing, like ConcertChat [5]. In such cases, some complications appear, because the conversation does not follow only one thread, multiple topics being discussed in parallel. Therefore, a new model is needed, which allows the understanding of the collaboration mechanisms and provides the means to measure the contributions of participants: the inter-animation and the polyphony theory identified by Bakhtin [2] which states that in any text there is a co-occurrence of several voices that gives birth to inter-animation and polyphony: "Any true understanding is dialogic in nature." [13]. The same idea is expressed in [8]: "knowledge is socially built through discourse and is preserved in linguistic artefacts whose meaning is co-constructed within group processes".

For the moment, there are very few systems that use the polyphony theory for the conversation's analysis, PolyCAFe [12] being one such example. This system analyzes the contribution of each user and provides abstraction and feedback services for supporting both learners and tutors. It uses Natural Language Processing techniques that allow the identification of the most important topics discussed (with TF-IDF and Latent Semantic Analysis), speech acts, adjacency pairs, Social Network Analysis in order to identify the conversation threads and the individual involvement of the participants.

In this paper, we present a system that also starts from Bakhtin's polyphony theory [2, 3], where by voice we understand either a participant to the chat, or an idea (a thread of words that are present throughout the chat in order to represent something). This larger view of the notion of "voice" was inspired by Tannen's ideas [11] related to the use of repetitions as a measure of involvement. The purpose of the system is to evaluate the quality of the whole conversation from the point of view of participants' involvement in the conversation and by the effectiveness of the conversation from some given key-concepts point of view.

The paper continues with the presentation of the functions of repetitions and the information that we have extracted from chat conversations considering these functions. After that, we present the results of the application's validation and what we have undertaken for its verification. The paper concludes with our final remarks.

2 Functions of Repetition

Deborah Tannen identified four major functions of repetitions in conversations: production, comprehension, connection and interaction. She also pointed out that these functions taken together provide another one – the establishment of coherence as interpersonal involvement [11].

Repetition "facilitates the production of more language, more fluently" [11]. People are supposed to think about the things that they are about to utter and using repetition, the dead times that could appear during this time are avoided, and thus the fluency of the talk is increased.

The comprehension benefits from the use of repetitions in two ways. First of all, the information is not so dense when using repetitions and the one receiving it has enough time to understand it. Secondly, repetition is also useful for comprehension because usually only the important concepts are repeated, which signals what is the real message of the conversation, or what does it emphasize.

The repetition also serves as a linking mechanism for connecting the phrases from the text. Through repetition, the transition between ideas is softer, and the topics seem to be better connected. Repetition "serves a referential and tying function" [4].

In the same time, repetition has a role in connecting the participants also, because the author is able to present his opinion on the spoken subjects, emphasizing the facts that he/she believes are of greater importance and trying to induce the same feelings in the audience. Therefore, the repetition also has an interactional role by bonding the "participants to the discourse to each other, linking individual speakers in a conversation and in relationships" [11].

According to Tannen [11], the combination of all the previous functions leads to a fifth purpose – the creation of interpersonal involvement. Repeating the words of the other speakers, one shows his/her response according to what previous speakers said, along with his/her attitude by presenting their own facts and therefore keeping the conversation open to new interventions.

Tannen considers that "dialogue combines with repetition to create rhythm. Dialogue is liminal between repetitions and images: like repetition is strongly sonorous" [11].

3 Extracted Information

Considering the above ideas, we have built an application that tracks the repetitions from a conversation and evaluates the contribution of the users in terms of their involvement and the quality of the conversation in terms of some given key concepts that needed to be debated. In this analysis, we did not consider repetition only as exact apparition of the same word, but in the broader sense of repetition of a concept determined using lexical chains built using WordNet (http://wordnet.princeton.edu).

The information that we collected was both qualitative and quantitative:

- how *interesting* is the conversation for the users - counted as the number of a user's replies, since once a conversation is interesting for a user, it is more likely that he/she will be interested in participating and therefore will utter more replies than if he/she is not interested in the subject debated in the conversation;
- *persistence* of the users -the total number of the user's consecutive replies;
- *explicit connections* between the users' words - considered as the explicit references made by the participants (facility provided by ConcertChat environment);
- *activity* of a user - the average number of uttered characters per reply for that user. This information is needed in addition to the number of uttered replies because we desire that the answers to be as elaborate as possible, thus giving a higher probability to the apparition of important concepts;

- *absence* of a user from the conversation - determined as the average time between a user's consecutive replies;
- *on topic* - a qualitative measure of the conversation, showing to what degree the participants used concepts related to the ones imposed for debating. This measure is intended to penalize the off-topic debate;
- *repetition* - how often a participant repeats the concepts introduced by others, showing the interaction between users and the degree of attention devoted by one participant to the words of the others;
- *usefulness* of a user - how often the concepts launched by a user have been used by a different participant;
- *topic rhythmicity* - the number of replies between two consecutive occurrences of the same topic. This measure is also intended to eliminate off-topic talk.

Once we decided what information will be extracted, we needed to determine the threshold values that allow us to consider a chat to be useful or not from the debated concepts and the participants' involvement points of view.

In order to determine these values, we considered 6 chats consisting of 1886 replies – ranging from 176 to 559 replies – that have been created by undergraduate students in the senior year involved in the Human-Computer Interaction (HCI) course using the ConcertChat environment [5]. They were divided in small groups of 4-5 students, every participant having to present a web-collaboration platform (chat, forum, blog, wiki) and prove its advantages over the ones chosen by the other participants.

The purpose of the chats was to facilitate the understanding of the pros and cons of all the given platforms and to find the best (combination of) communication and collaboration technologies to be used by a company to support team work.

These chats were automatically analyzed using the application that we have developed. A couple of tests have been developed starting from the collected information. For each of these tests, the application gives a grade from 0 to 10, specifies if that test have been passed or not and what was the cause of that test (a person, a topic or an overall criterion). See Figure 1 for an output of the application.

Based on the obtained values we identified the thresholds and the tendencies to be desired for a chat. All these values are presented in Table 1.

As it can be seen, we usually want high values: we want both the most and least interested person in the chat to be as active as possible (test 1 and 2) because it means they had a reason to discuss more – either for presenting more information or for debating more the given topics; we want the explicit connections between users to be as high as possible (thus showing the involvement in the discussion – test 4); the minimum/maximum activity (reflected as the average number of words/characters per utterance) should be high as well, because we desire to have elaborated sentences and not just key words appearing sporadically (test 6 and 7); the chat should contain as many words as possible related to the subjects given as an input, therefore discouraging spamming and off-topic discussions (test 8); we also should look for high values in repetitions, for they tell us that the users were paying attention to other users' words (tests 9 and 10); and finally, we want users to say important things that can be useful for the other participants to better understand the debated subjects and that can help them build their own ideas on those users' words (tests 11 and 12).

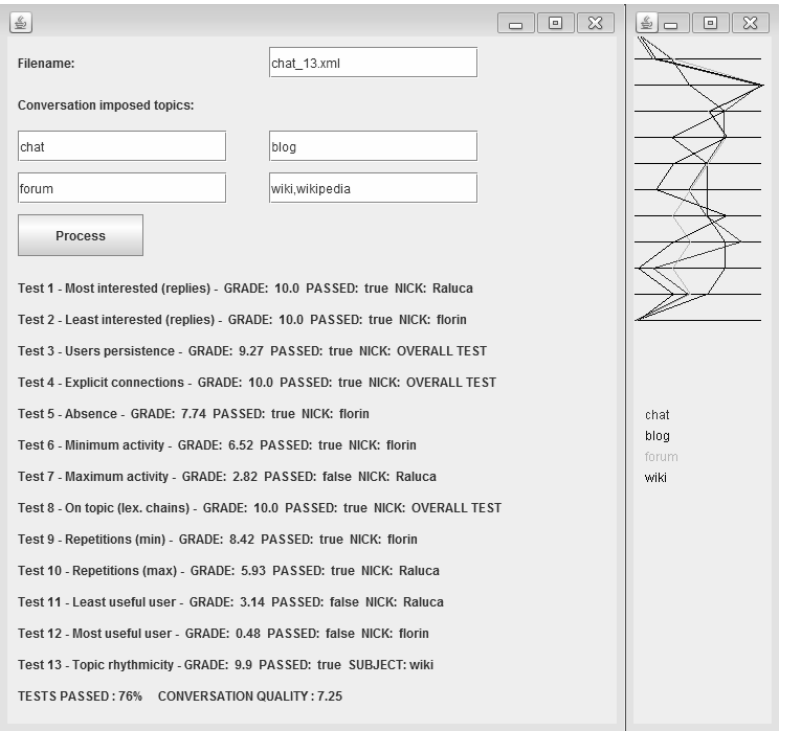


Fig. 1. Application output for a given chat

Table 1. The values obtained for the chats with 4-5 participants involved and the desired tendencies for the tests

Name of test	Tendency	Chat 1	Chat 2	Chat 3	Chat 4	Chat 5	Chat 6	Min	Max
0. Utterance number	high	183	291	377	176	559	300	176	559
1. Most interested	high	0.38	0.4	0.3	0.34	0.38	0.32	0.3	0.4
2. Least interested	high	0.11	0.18	0.15	0.16	0.12	0.08	0.08	0.18
3. Users persistence	low	0.31	0.08	0.16	0.07	0.17	0.17	0.07	0.31
4. Explicit connections	high	0.43	0.79	0.27	0.4	0.19	0.48	0.19	0.79
5. Absence	low	0.047	0.019	0.016	0.033	0.011	0.037	0.011	0.047
6. Minimum activity	high	8	36	27	14	52	6	6	52
7. Maximum activity	high	59	132	93	48	345	48	48	345
8. On topic - lex. chain	high	0.23	0.28	0.21	0.19	0.2	0.25	0.19	0.28
9. Repetitions (min)	high	0.1	0.13	0.1	0.15	0.08	0.11	0.08	0.15
10. Repetitions (max)	high	0.14	0.15	0.13	0.18	0.12	0.15	0.12	0.18
11. Least useful user	high	1.84	2.03	2.1	1.73	2.1	2.69	1.73	2.69
12. Most useful user	high	2.12	2.16	2.16	2.16	2.36	2.95	2.12	2.95
13. Topic rhythmicity	low	1.36	0.76	1.51	1.69	0.92	1.22	0.76	1.69
14. Passed tests (%)	high	15	76	23	30	46	53	15	76
15. Quality	high	2.72	7.25	3.44	4.1	5.08	4.62	2.72	7.25

Now we shall focus our attention on the tests where small values are required. The first test that shows such a characteristic (test 3) is related to the users' persistence in the chat expressed as the number of consecutive replies uttered by a user without the intervention of the other participants, and based on our results we want them to be as few as possible. The idea behind this is the fact that too many consecutive replies of the same user show that the other participants had nothing to add or comment and that is a sign of not being involved, not paying attention to what that user had to say. More than that, when a user utters too much content that is not interesting for the other participants, they tend to get bored and they lose the interest in the conversation as a whole, which results in even less intervention from their part and a poor quality conversation. The second test that requires small values to show a high involvement of the participants is test number 5, which measures the maximum time between two consecutive replies of a user. If a user is taking too long to respond then he/she is not actively participating in that chat (the user is considered to be missing that part of the conversation). The last test needing small values is test number 13, which basically states that we need a small number of replies between two consecutive occurrences of a specific topic – the given topics should have high frequencies in the conversation. We desire a constant deliberation on all topics and not just users speaking in turns about the topics that were provided in order to be debated. This test also has a graphical representation of the provided topics' rhythmicity, therefore it is easier to understand what we measure, based on its graphical depiction (see Figure 2).

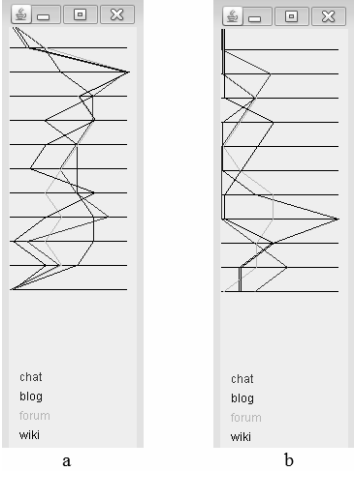


Fig. 2. Graphical representation of topic rhythmicity. a) chat with high rhythmicity for the debated topics; b) chat with poor rhythmicity for some of the topics.

In the above figure, there are two examples of rhythmicity in chats. The chats have been divided in equal shares (separated by the horizontal lines) and, in each of them, the topics to be debated are represented by a different line. The more a topic is debated in a share of a conversation, the closer is the line representing that topic to the right side of that share of the chat graphical representation. Figure 2.a. shows a chat

with high rhythmicity for all topics – these were debated in parallel as it can be seen by the lack of flat lines near the left side of the representation. The other figure (2.b.) shows the opposite: it has flat lines on the left side of the graphic showing that the topic that they represent has not been debated in those parts of the chat. The conversation starts with a discussion about blogs, while the other topics are ignored. As time passes, these topics get into focus in the detriment of chat, which seems to be forgotten for a while (it is absent in three of the eleven shares of the given chat). The end of the conversation finds all the given topics in focus, as it is desirable, but having long periods of one-topic debate – the topics have been debated in turns which means the participants did not compare them and therefore did not achieve one of the purposes of the conversation.

Test 14 shows the percentage of the passed tests (tests where the obtained grade was above 5) considering the min and max as inferior and superior thresholds, while test 15 represents the average grade obtained by the chat for the 13 tests.

4 Validation

First of all we needed to validate the results obtained with the application and therefore we asked two HCI teachers to evaluate the chats having in mind two main criteria: the quality of the content related to the given concepts and the participants' involvement. Their grades, along with the average values and the scores provided by our application are presented in Table 2.

Table 2. The gold standard values provided for the 6 chats along with the scores provided by our application and with the revised values

Chat	Reviewer 1	Reviewer 2	Average	Application score	Modified app. Score
Chat 1	7.8	7.74	7.77	2.72	7.08
Chat 2	10	9.3	9.65	7.25	10
Chat 3	9	8.9	8.95	3.44	7.8
Chat 4	8.4	8.6	8.5	4.1	8.46
Chat 5	10	9	9.5	5.08	9.44
Chat 6	9	9	9	4.62	8.98

As it can be easily seen, the application's grades are much smaller than the ones provided by the reviewers and therefore we increased these grades by the average of the difference between the reviewers' grades and the scores provided by the application (4.36). The new values are presented in Figure 3 below.

Before modifying the application scores, we had to see how trustworthy were the grades provided by the reviewers and therefore we computed their correlation. This value was 0.8829, which shows that their values are very similar and being domain experts and having experience in teaching, we decided we can trust the values provided. We have also computed the correlation between the reviewers' average grades and the scores provided by the application. The value was 0.8389, very close to the correlation between the reviewers, showing a strong correlation between the application's grades and the real value of the chats.

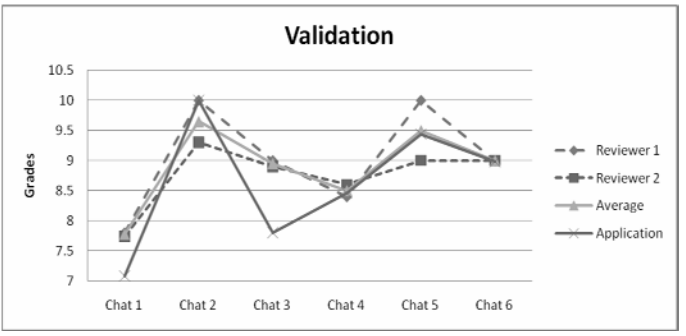


Fig. 3. Application's validation

5 Verification of the Model

We considered two different verification methods for our application. The first one was meant to demonstrate that the model used for a chat conversation depends very much on the number of participants. Therefore, we considered 4 chats consisting of 1250 replies that had between 6 and 8 participants. These chats had the same focus and objectives as the ones used for the application's validation. These chats have been automatically analyzed using our application in order to see whether the model for 4-5 participants could have been also applied for them. The values obtained, along with the thresholds for these chats and for the chats having 4-5 participants are presented in Table 3. The results clearly show that the model for 4-5 participants (represented by the used thresholds) is not adequate for chats with 6-8 participants.

Table 3. Differences between chats with 4-5 participants (chats 1-6) and chats with 6-8 participants

Name of test	Chat 7	Chat 8	Chat 9	Chat 10	Min 6-8	Max 6-8	Min 4-5	Max 4-5
0. Utterance number	138	380	473	259	138	473	176	559
1. Most interested	0.31	0.27	0.21	0.19	0.19	0.31	0.3	0.4
2. Least interested	0.06	0.04	0.07	0.02	0.02	0.07	0.08	0.18
3. Users persistence	0.3	0.02	0.004	0.003	0.003	0.3	0.07	0.31
4. Explicit connections	1.04	1.01	1.01	1.03	1.01	1.04	0.19	0.79
5. Absence	0.12	0.058	0.028	0.134	0.134	0.028	0.011	0.047
6. Minimum activity	1	2	16	1	1	16	6	52
7. Maximum activity	21	137	90	46	21	137	48	345
8. On topic - lex. chain	0.2	0.19	0.29	0.34	0.19	0.34	0.19	0.28
9. Repetitions (min)	0.07	0.03	0.06	0.06	0.03	0.07	0.08	0.15
10. Repetitions (max)	0.13	0.09	0.09	0.13	0.09	0.13	0.12	0.18
11. Least useful user	2.77	3.53	4.24	4.09	2.77	4.09	1.73	2.69
12. Most useful user	3.39	4.96	5.1	4.81	3.39	5.1	2.12	2.95
13. Topic rhythmicity	1.54	1.46	0.51	0.69	0.51	1.46	0.76	1.69

After we have seen that the thresholds do not match, we wanted to verify that we have the same type of chats as previously presented. Consequently, we have modified these chats by considering not the physical participants, but the point of view - "the voice" - that they represent. Therefore, we considered the persons debating the same topics as being a single participant and thus we ended up having again chats with 4 participants debating the same topics as before. These chats have been automatically evaluated and the results showed that they fit well enough in the model with only 4-5 participants, as it can be seen in Table 4. In conclusion, the chats were not different from what we have seen already, but the thresholds were not adequate for them.

The second, and maybe the most important verification method, was tested on three different chats from the same set with the ones used for learning and validation (4 participants debating about chat, forum, blog and wiki), consisting of 911 replies, and asked another teacher of the HCI class to evaluate them in the same fashion as for validation. After that, the chats have been automatically evaluated using our application and the correlation between the reviewer and the application's grades has been computed. The correlation was 0.7933. The values are presented in Table 5.

Table 4. The values obtained for chats 7-10 modified to have 4 participants

Test No.	Mod 7	Mod 8	Mod 9	Mod 10	Min mod	Max mod	Min 6-8	Max 6-8	Min 4-5	Max 4-5
Test 1	0.42	0.35	0.28	0.3	0.28	0.42	0.19	0.31	0.3	0.4
Test 2	0.13	0.13	0.22	0.19	0.13	0.22	0.02	0.07	0.08	0.18
Test 3	0.17	0.1	0.05	0.02	0.02	0.17	0.003	0.3	0.07	0.31
Test 4	1.02	1.01	1	1.01	1	1.02	1.01	1.04	0.19	0.79
Test 5	0.055	0.018	0.009	0.02	0.009	0.055	0.134	0.028	0.011	0.047
Test 6	4	30	106	60	4	106	1	16	6	52
Test 7	41	232	188	115	41	232	21	137	48	345
Test 8	0.2	0.2	0.29	0.34	0.2	0.34	0.19	0.34	0.19	0.28
Test 9	0.13	0.1	0.11	0.17	0.5	1.53	0.03	0.07	0.08	0.15
Test 10	0.18	0.136	0.12	0.19	0.12	0.19	0.09	0.13	0.12	0.18
Test 11	1.88	1.98	2.01	1.91	0.1	0.17	2.77	4.09	1.73	2.69
Test 12	2.14	2.36	2.38	2.14	1.88	2.01	3.39	5.1	2.12	2.95
Test 13	1.53	1.49	0.5	0.68	2.14	2.38	0.51	1.46	0.76	1.69

Table 5. The gold standard values provided for the 3 chats along with the scores computed by our application and with the revised values

Chat	Reviewer	Application	Modified application score
Chat 11	9.627	5.24	9.6
Chat 12	7.574	4.76	9.12
Chat 13	8.777	5.39	9.75

6 Conclusion and Further Work

In this paper we have presented an application that evaluates the quality of a chat according to a number of predefined conversation topics and to the personal involvement

of the participants. During the verification, we have shown that the models that should be used to evaluate the chats are dependent on the number of participants: they are different for small (4-5 participants) and medium (6-8 participants) teams, and we expect that these models are also different for 2-3 participants and for more than 8 participants.

The good correlation between the application and the domain experts obtained at both the validation and verification stages recommends it as a reliable application. Also, the large number of tests, gives a lot of flexibility to the user, allowing him/her to give more or less importance to some of the tests and therefore to evaluate exactly the aspects considered to be important.

In the meantime, an evaluator can make a complex analysis of the chats by correlating the results of the different tests, this way identifying the causes that lead to the obtained results and thus being able to take the right decision in the evaluation.

Acknowledgments. This research presented in this paper was supported by project No.264207, ERRIC-Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1.

References

1. Adams, P.H., Martell, C.H.: Topic detection and extraction in chat. In: Proceedings of the 2008 IEEE International Conference on Semantic Computing, pp. 581–588 (2008)
2. Bakhtin, M.M.: Problems of Dostoevsky's Poetics. Ardis (1993)
3. Bakhtin, M.M.: The Dialogic Imagination: Four Essays. University of Texas Press (1981)
4. Halliday, M.A.K., Hasan, R.: Cohesion in English. Longman, London (1976)
5. Holmer, T., Kienle, A., Wessner, M.: Explicit referencing in learning chats: Needs and acceptance. In: Nejdl, W., Tochtermann, K. (eds.) EC-TEL 2006. LNCS, vol. 4227, pp. 170–184. Springer, Heidelberg (2006)
6. Jurafsky, D., Martin, J.H.: Speech and Language Processing. In: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd edn. Pearson Prentice Hall, London (2009)
7. Kontostathis, A., Edwards, L., Bayzick, J., McGhee, I., Leatherman, A., Moore, K.: Comparison of Rule-based to Human Analysis of Chat Logs. In: Conferencia de la Asociación Española para la Inteligencia Artificial (2009)
8. Rebedea, T., Trausan-Matu, S., Chiru, C.-G.: Extraction of Socio-semantic Data from Chat Conversations in Collaborative Learning Communities. In: Dillenbourg, P., Specht, M. (eds.) EC-TEL 2008. LNCS, vol. 5192, pp. 366–377. Springer, Heidelberg (2008)
9. Schmidt, A.P., Stone, T.K.M.: Detection of topic change in IRC chat logs, http://www.trevorstone.org/school/ircsegmentation.pdf
10. Sfard, A.: On reform movement and the limits of mathematical discourse. Mathematical Thinking and Learning 2(3), 157–189 (2000)
11. Tannen, D.: Talking Voices: Repetition, Dialogue, and Imagery in Conversational Discourse. Cambridge University Press, Cambridge (1989)
12. Trausan-Matu, S., Rebedea, T.: A polyphonic model and system for inter-animation analysis in chat conversations with multiple participants. In: Gelbukh, A. (ed.) CICLing 2010. LNCS, vol. 6008, pp. 354–363. Springer, Heidelberg (2010)
13. Voloshinov, V.N.: Marxism and the Philosophy of Language. New York Seminar Press (1973)

Emotion Based Music Visualization System

Jacek Grekow

Faculty of Computer Science, Bialystok University of Technology,
Wiejska 45A, Bialystok 15-351, Poland
j.grekowj@pb.edu.pl

Abstract. This paper presents a new strategy of visualizing music. The presented method is a transformation of musical content in dynamically changing spatial figures. The proposed visualization system was linked with an external system of automatic emotion detection. The labels produced by the system were used to modify certain parameters of the visualization. Therefore, because the visual presentation of emotions can be subjective, the system was expanded with user profiles obtained by grouping questionnaires. This allowed the adjustment of some parameters of visualization to user preferences.

Keywords: Music visualization, harmonic analysis, consonance and dissonance, emotion detection.

1 Introduction

At a time when live musical concerts often have scene decoration, flashing lights and choreographed dance moves, just listening to a piece of music from the speakers does not always provide sufficient satisfaction. Many listeners are accustomed to video clips in which the music is accompanied by attractive images, photos, videos, etc. The system presented in this paper automatically generates 3-D visualizations based on MIDI files. Spatial figures created in real-time additionally enrich the music experience, allowing for a perception of music through an additional communication channel so that this perception becomes richer and fuller.

1.1 Previous Work

There are different attitudes towards the visualization of music and it still is not a fully solved problem. The attitudes focus on the different elements of music such as pitch, rhythm, dynamics, harmony, timbre or attempt a comprehensive analysis of the forms of musical composition [1], [2]. Smith and Williams [3] use colored spheres to visualize music, which they place in 3-D space. The characteristics of these visual objects depend on properties that describe musical tones: pitch, volume and timbre. Foote [4] presented a method for visualizing the structure of music by its acoustic similarity or dissimilarity in time. Acoustic similarity between two instants of an audio recording was calculated and presented in a two-dimensional representation. While constructing a visualization, Whitney [5]

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 523–532, 2011.

refers to the ancient principles of harmony created by Pythagoras, based on the relations of natural numbers. He tries to find a relationship between the harmony in sound and its visualization. To present musical consonance and dissonance in visual form, he uses points moving in space whose speed is dependent on the numerical relations of simultaneously sounding tones. This work has been continued by Alves [6]. Bergstrom, Karahalios and Hart [7] focus on the analysis of the harmonic aspects of music. With the help of figures called isochords, they visualize musical chords in 2-D space (Tonnetz coordinate system), which emphasizes consonant intervals and chords. Ciuha, Klemenc and Solina [8] also made an attempt to visualize the harmonic relationships between tones. The visualization uses a three-dimensional piano roll notation, in which the color depends on the simultaneously sounding tones. Consonant tone combinations are represented by saturated colors and dissonant are unsaturated. Isaacson put forth a summary of selected music visualization methods presenting both content and sound files as well as files used for the analysis of musical form [9].

1.2 System Construction

The presented visualization system (Fig. 1) was linked with an external system of automatic emotion detection [10]. The labels produced by the external system were used to modify certain parameters of the visualization. When using the system, the user first completes a questionnaire through which the system assigns the user to one of the profiles. The user then selects a MIDI file to be played and visualized. The visualization module uses an external system for automatic emotion detection in the MIDI file. The discovered emotions partially affect the created visualizations by changing the parameters of the drawn spatial figures.

The emotion labels obtained from the external system for automatic emotion detection are classified into four main groups corresponding to four quarters of the Thayer model [11]: $e1$ (energetic-positive), $e2$ (energetic-negative), $e3$ (calm-negative), $e4$ (calm-positive).

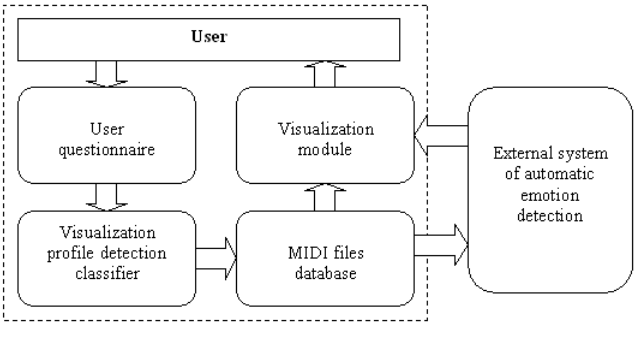


Fig. 1. The construction of the personalized music visualization system

2 Method of Creating Spatial Figures

The harmonic content of the composition consists of chords assigned to the successive sections of the musical piece. Let's examine a chord consisting of three sounds. Because every musical tone can be presented - to simplify - as an ideal sinusoid of frequency f, we describe each component of the chord Ak sinusoidal function $S_i(t)$ with frequency f_i. Assigning each of the obtained functions one of the X, Y and Z axis of the Cartesian system $\{U\}$, we build a spatial figure Φ, corresponding to the Ak chord:

$$Ak \rightarrow \Phi \tag{1}$$

For a thus obtained figure we assume the name: AKWET = AKkordWertETalon. The name AKWET was created by the juxtaposition of the following words: chord (german Akkord), value (german Wert) and pattern (french Etalon).

It is possible to describe the formation of each AKWET as a result of movement of point P whose spatial position is determined by the values of function $S_i(t)$ placed respectively on the X, Y and Z axis (Fig. 2).

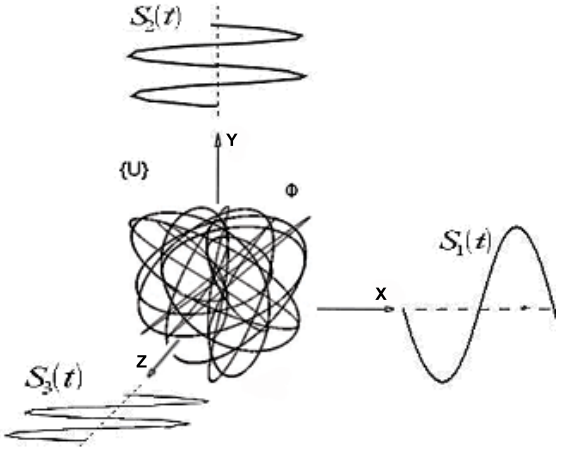


Fig. 2. Creating an image of an exemplary musical pattern

We assign to each axis of the system $\{U\}$ (X, Y and Z) a sinusoidal signal $S_i(t)$ with frequency f_i, where $i = 1, 2, 3$. These signals reflect different components of the pattern. If we simultaneously subdue signals $S_i(t)$ to discretization with sampling frequency $F \gg f_i$, we obtain a sequence of samples whose every element we consider as a ternary vector, determining the position of point P_j. The coordinates of a single point P_j, which are calculated on the basis of a joint time t and the function of $S_i(t)$ signals, can be noted as follows:

$$P_j = (P_{jx}, P_{jy}, P_{jz}) \tag{2}$$

$$P_{jx} = S_1(t_j) = A_1 sin\omega_1 t_j \tag{3}$$

$$P_{jy} = S_2(t_j) = A_2 sin\omega_2 t_j \tag{4}$$

$$P_{jz} = S_3(t_j) = A_3 sin\omega_3 t_j \tag{5}$$

AKWET Φ is a spatial figure occurring in time (Fig. 2), and its duration depends on the value of the component signals. The construction of an AKWET is based on a principle similar to that on which the Lissajous figures are constructed [12]. But the important difference in this approach is that: firstly, the obtained figures are spatial, and secondly, they are used to visualize and analyze harmonic musical content.

3 Figure Parameters

The parameters describing the formed spatial figures (AKWETs) were divided into two groups. The first consists of drawing parameters that were permanently defined by the designer. The second group consists of parameters that are modified on the basis of the emotion obtained from the external system for emotion detection.

3.1 Unchanging Figure Parameters

Form of the Figures. The form of the figures is closely related to the method of figure formation and presents closed lines drawn in 3-D space, the complexity of which depends on the chord's dissonance ability. Below is a visualization of two basic chords: major and minor, whose sound is significantly different and simply said: the major chord is happy and the minor sad [13].

It turns out that the figures corresponding to the major and minor chords differ substantially. The form of the major chord (Fig. 3a) is much simpler, cleaner and transparent. Whereas, the minor chord (Fig. 3b) creates a complicated, multilayered form, and despite a proper shape, it is difficult to associate it with calm.

Figure Sizes. In the proposed system of sound consonance visualizations, the size of the figures is dependent on the volume of the music. The louder the volume, the larger the figures are drawn.

Below are two chords of the same harmonic content (major triad) but of a different volume of the components making up the chords. The first one is quieter (Fig. 4a), and the second louder (Fig. 4b). Different volumes of the individual components of the chord caused not only a change in the size of the figures, but also stretching and narrowing of its shape.

Number of Figures Shown Simultaneously. The more sounds in harmony at the same time, the richer the harmonic sound that is obtained. Each new component of harmony enters into a relationship with the others. This is expressed in the number of figures constructed during the visualization (Fig. 5).

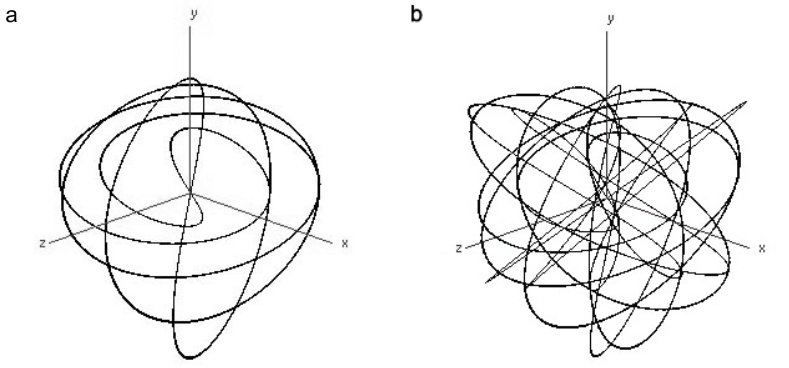


Fig. 3. AKWET from a major chord (a) and a minor chord (b)

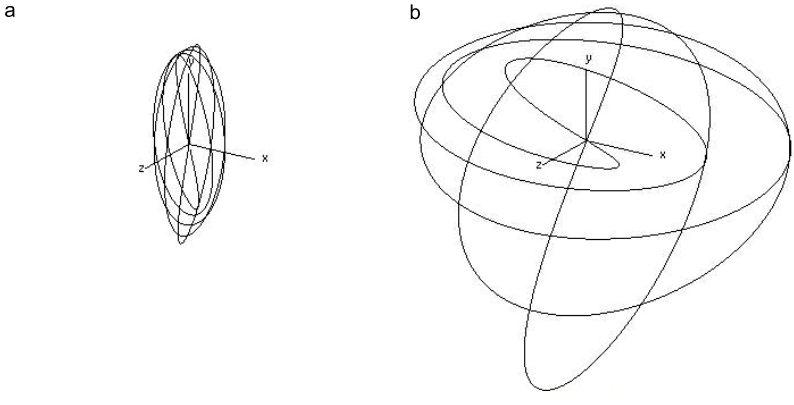


Fig. 4. AKWET created from a quiet major triad (a) and from a loud major triad (b)

Until there are less than 4 audio components, a single AKWET is used. When the amount of harmonious sounds increases, the number of AKWETs increases as well, and their number is expressed as the amount of 3-voice combinations without repetitions.

3.2 Figure Parameters Modified According to the Detected Emotion

To emphasize the way the figures are drawn, depending on the emotion detected by the external system for automatic emotion detection, it was decided to change the parameters of the drawn line (thickness and type). Three types of line thicknesses were selected: thin, medium, thick, and three types of line shapes: continuous, dashed and dotted. The change of line type affected the visual expression of the AKWETs formed.

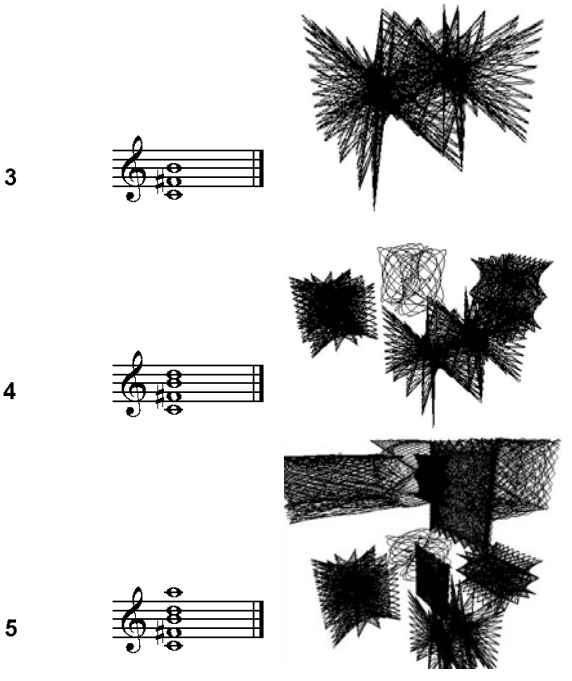


Fig. 5. Number of tones forming a harmony and type of visualization

4 Tester Questionnaires

The first step in the process of user profile construction was the collection of questionnaires from the observer-listeners. The listeners came from different professional groups (students, musicians, scientists, engineers, teachers, medical workers) and different age groups (from 20 to 50 years old). The study was conducted on 41 people who filled out the *visualization questionnaire*, in which they defined the parameters of the line that according to them should be used to draw figures visualizing music with a particular emotion. For each emotion ($e1$, $e2$, $e3$, $e4$), a tester chose one line type: continuous, dashed, dotted, and one thickness: thin, medium and thick. For example, one of the choices might look like this: for $e1$ (energetic-positive thin and continuous line, for $e2$ (energetic-negative) medium and dashed line, etc.

Each of the testers also completed an additional *personality questionnaire*, which was used to construct the user profiles. The set of questions used in this questionnaire was as follows: Sex; Profession; Basic education; Age; How do you feel at the moment?; Do you like reading books?; What are your hobbies?; What kind of music do you prefer?; What is you favorite instrument?; What are your two favorite ice cream flavors?; If you had $3,500.00 what would you most likely spend it on?; How big is the city you live in?; Are you happy with your life and

what you have achieved so far?; Do you consider yourself to be a calm person?; Do you live with your family?; Does your job give you pleasure?; Do you like playing with and taking care of little children?; Do you have or would you like to have a little dog at home?

5 The Experiment Results

5.1 Clustering and Searching for User Profiles

To search for groups and classifications, the set of tools provided in the WEKA package was used [14]. The *visualization questionnaires* were grouped with the help of algorithms (K-Means and Expectation maximization). In each group, there were testers who similarly answered the question which line should be used to draw a certain emotion.

The resulting group and the responses from the *personality questionnaires* were used to find the characteristics of these groups. A decision table was created, in which the rows were the testers, and their answers to the questions from the *personality questionnaire* were the columns. The last column was the group found during grouping of the *visualization questionnaires*. This way, the created data were classified using the following algorithms: PART, JRip, J48, and the models that best described the collected data were found. An additional selection of attributes was carried out using WrapperSubsetEval [15], which was tested with the following algorithms: PART, J48, k-NN. This way, the most significant questions describing the profiles - groups were found.

The winner of the classification was algorithm J48 (80% Correctly Classified Instances) and the most appropriate set of grouping was the set obtained by using K-Means (number of clusters equals 2).

5.2 Group Characteristics

From the selected attributes, it turned out that the most meaningful questions and answers from the *personality questionnaire* were the questions presented in Table 1.

Table 1. The most meaningful questions and answers in establishing the user profile

Question	Answer
Age	20-30
Do you consider yourself to be a calm person?	Yes

Explaining the meaning of the questions chosen in the selection process of attributes gives rise to the conclusion that people who consider themselves to be calm would like to draw figures corresponding to the emotions differently than the people who do not consider themselves to be calm. Age also proved to be significant in this regard. Below (Table 2) are descriptions of groups that were read from the structure of the tree constructed by the J48 algorithm.

Table 2. Group characteristics

Group	Number of people	Description
Cluster 1	26	People who: - are aged 20-30 and consider themselves to be calm
Cluster 2	15	People who: - are over 30 years of age, - if they are aged 20-30, they do not consider themselves to be calm

Table 3. Mapping the type of emotion to the type of line

Type of emotion	Centroid 1	Centroid 2
$e1$	continuous line, medium	dotted line, thick
$e2$	dashed line, thick	continuous line, thick
$e3$	dotted line, thin	dashed line, thick
$e4$	continuous line, thin	continuous line, thin

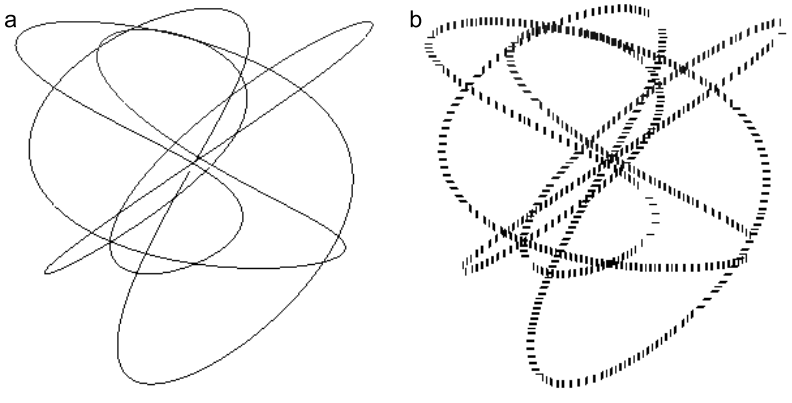


Fig. 6. Visualization of the chord during emotion $e1$ for profile 1 (a) and the chord during emotion $e1$ for profile 2 (b)

5.3 Profile-Dependent Visualization

During the grouping, algorithm K-Means was also able to find the representatives of groups: centroids. The line parameters of group representatives (Table 3) were used during the visualization. Depending on the assigned user profile, the appropriate type of line to draw the AKWETs was chosen.

From the observation of centroids, it seems that the type of line (thin and continuous) is the same for emotions $e4$, while the lines are different for all the other emotions. A representative of group 2 (Centroid 2) uses thick lines a lot more often (emotions $e1$, $e2$, $e3$). Interestingly, both representatives of the groups chose a thick line for emotion $e2$ (energetic-negative).

Presented below is the same chord during emotion $e1$ (energetic-positive). Depending on the identified system user profile, we obtain two different visualizations. For the first profile a continuous and thin line was used (Fig. 6a), and for the second profile a dotted and thick line (Fig. 6b). Using different lines also changes the character of the figure. The figure drawn by the continuous line is drawn subjectively calmer than the dotted line. The figure drawn by the thick line is less noble than the one drawn with a thin line.

6 Conclusion

This paper presents a new strategy of visualizing music. The presented method is a transformation of musical content in dynamically changing spatial figures. The proposed visualization system was linked with an external system of automatic emotion detection. The labels produced by the system were used to modify certain parameters of the visualization. Mapping the emotion labels with the parameter of drawing figures additionally strengthened their expression and attractiveness. Therefore, because the visual presentation of emotions can be subjective, the system was expanded with user profiles obtained by grouping questionnaires. This allowed the adjustment of some parameters of visualization to user preferences. The number of visualization parameters dependent on the emotions are not closed and may be expanded in the future.

Acknowledgments. This paper is supported by the S/WI/5/08.

References

1. Sapp, C.: Visual Hierarchical Key Analysis. ACM Computers in Entertainment 4(4) (2005)
2. Wu, H., Bello, J.P.: Audio-based music visualization for music structure analysis. In: Sound and Music Computing Conference, Barcelona (2010)
3. Smith, S.M., Williams, G.N.: A visualization of music. In: VIS 1997: Proceedings of the 8th Conference on Visualization 1997, pp. 499–503 (1997)
4. Foote, J.: Visualizing music and audio using self-similarity. In: ACM Multimedia 1999, pp. 77–80 (1999)
5. Whitney, J.: Digital Harmony: On the Complementarity of Music and Visual Art. Byte Books, New York (1980)
6. Alves, B.: Digital Harmony of Sound and Light. Computer Music Journal 29(4) (Winter 2005)
7. Bergstrom, T., Karahalios, K., Hart, J.C.: Isochords: Visualizing structure in music. In: GI 2007: Proceedings of Graphics Interface 2007, pp. 297–304 (2007)
8. Ciuha, P., Klemenc, B., Solina, F.: Visualization of concurrent tones in music with colours. In: MM 2010, Italy (2010)
9. Isaacson, E.J.: What you see is what you get: on visualizing music. In: ISMIR, pp. 389–395 (2005)
10. Grekow, J., Raś, Z.W.: Detecting emotions in classical music from MIDI files. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 261–270. Springer, Heidelberg (2009)

11. Thayer, R.E.: The biopsychology arousal. Oxford University Press, Oxford (1989)
12. Shatalov, M.I.: Calculation of errors in comparing frequencies by means of Lissajous figures. Measurement Techniques 2(3), 215–218 (1959)
13. Sikorski, K.: Harmonia. Polskie Wydawnictwo Muzyczne, Krakow (1965)
14. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)
15. Hall, M.A., Holmes, G.: Benchmarking Attribute Selection Techniques for Discrete Class Data Mining. IEEE Transactions on Knowledge and Data Engineering 15(6), 1437–1447 (2003)

Notes on Automatic Music Conversions*

Wladyslaw Homenda[1,2] and Tomasz Sitarek[1]

[1] Faculty of Mathematics and Information Science, Warsaw University of Technology
Plac Politechniki 1, 00-660 Warsaw, Poland
[2] Faculty of Mathematics and Computer Science, University of Bialystok
ul. Sosnowa 64, 15-887 Bialystok, Poland

Abstract. Aspects of automatic conversion between sheet music and Braille music are studied in this paper. The discussion is focused on syntactical structuring of information carried as sheet music and as Braille music (their digital representations). The structuring is founded on context-free methods. Syntactical structures of a given piece of music create so called lexicon. Syntactical structures are reflected in the space of sounds/notes. This dependency is a many-to-one mapping for sheet music and is a many-to-many relation in case of Braille music. The dependency between lexicon and the space of sounds/notes defines semantics of the given language. Both syntactic structuring and semantic valuation produce a basis for an automatic conversion between both languages: printed music notation and Braille music.

Keywords: syntactical structuring, semantics, data understanding, music representation.

1 Introduction

Information exchange between human beings has been done in *languages of natural communication*, which are human languages (i.e. created and developed in a natural way by humans). Natural languages (English, Spanish, etc.) are the most representative examples of languages of natural communication. In this paper we focus the attention on processing music information. A special concern is given to printed music notation, also called sheet music, and Braille music description. Music description outlined as sheet music or as Braille music are tools used in inter-personal communication. Both languages of music description: printed music notation and Braille music satisfy the main feature of languages of natural communication: they have been created, developed and used prior to their formal codification and - up to now - they are not fully formalized. This is why we consider printed music notation and Braille music as languages of natural communication.

Our interest is focused on conversions between printed music notation and Braille music. The conversion to Braille music is an important social task, which

* This work is supported by The National Center for Research and Development, Grant no N R02 0019 06/2009.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 533–542, 2011.

helps to collect Braille music libraries for blind people. Such a conversion is difficult (time consuming, boring, error engendering etc.) for humans. It is a perfect task for automation. Lack of full formalization and codification of both languages (printed music notation and Braille music) raises difficulties in automation of such a task. Successful realization of conversions between both languages requires involvement of data structuring, information processing, domain knowledge etc. In the paper we discuss methods of information processing fundamental for these conversions: syntactical structuring and semantics, c.f. [3]. These methods of information processing are integrated in frames of the paradigm of granular computing, c.f. [1,4]. However, space of the paper does not allow for immersion of conversions in this paradigm.

The paper is organized as follows. In Section 2 we study syntactical structuring of music information carried in both languages of music description. Section 3 is dedicated to valuation of syntactical structures. The valuation defines semantics of languages. In Section 4 conversions are commented.

2 Syntactical Structuring

Syntactical approach to processing languages of natural communication (natural languages, music notation, etc.) is the study of how elementary items (words, notes) fit together to form structures up to the level of a complete constructions of a language (a sentence, a score). Syntactical approach is a crucial stage and a crucial problem in processing languages of natural communication.

Automatic analysis of languages of natural communication is commonly applied for natural languages. For several reasons automatic analysis of music notation has been a research niche not exploited intensively. In this study an effort is focused on syntactical description of music notation and Braille music. Then, syntactical constructions are immersed in a space of real objects described by syntactical constructions. Roughly, sounds played by a musical instruments are real objects.

The trouble with this approach is that human language is quite messy and anarchic, by comparison with languages of formal communication (like programming languages or MIDI format). Languages of formal communication are designed to conform to rigid (and fairly simple) rules. Such languages are almost exclusively used in communication with computers. Language constructions that break the rules are rejected by the computer. But it isn't clear that a language like music notation (or a natural language, English, Spanish) is rule-governed in the same rigid way. If there is a definite set of rules specifying all and only the valid constructions, the rules are certainly much more complicated than those formal languages. But, it seems that complexity is not the whole point - it is questionable whether a language of natural communication is fully definable by rules at all, and such rules as there are will often be broken with no bad consequences for communication. Thus, syntactical analysis of natural communication must be intensely flexible and deeply tolerant to natural anarchy of its subjects. This observations concerns, of course, music notation and Braille music. With

these notes in mind, the proposed approach to syntactical structuring will rely on sensible application of proposed context free grammars, i.e. that it will not be applied maliciously to generate incorrect examples.

2.1 Printed Music Notation

Below a method of construction of a context-free grammar $G = (V, T, P, S)$ to describe printed music notation is provided. In fact, the Table includes the set P of productions. The grammar presented in this Table shows a raw description of the printed music notation presented in Figure 1, an excerpt of Chopin's Grand Valse Brillante in E-flat Major, Op. 18. The set V of nonterminals includes all symbols in triangle brackets. The symbols $<score>$ stands for the initial symbols of the grammar S. The set T of terminals includes all names of symbols of music notation as well as symbols of music notation themselves. For the sake of clarity only basic music notation symbols are included with selected features, e.g. duration of eight and shorter notes are indirectly defined by beams and flags associated with the stem, while longer durations and pitches are not outlined at all. However, a way of expansion of the grammar to a broader set of symbols and properties of symbols is direct.

$<score>$ $\rightarrow$ $<system><score>$ | $<system>$
$<system>$ $\rightarrow$ $<part\_name><stave><system>$ | $<part\_name><stave>$
$<part\_name>$ $\rightarrow$ ε | Flauto | Oboe | Clarinet I | etc.
$<stave>$ $\rightarrow$ $<beg\_barline><bl\_stave>$ | $<bl\_stave>$
$<beg\_barline>$ $\rightarrow$ barline | etc.
$<bl\_stave>$ $\rightarrow$ $<clef><cl\_stave>$ | $<cl\_stave>$
$<clef>$ $\rightarrow$ treble_clef | bass_clef | etc.
$<cl\_stave>$ $\rightarrow$ $<key\_signature><ks\_stave>$ | $<ks\_stave>$
$<key\_signature>$ $\rightarrow$ ♯ | ♭ | ♯♯ | ♭♭ | ♯♯♯ | ♭♭♭ | *etc.*
$<ks\_stave>$ $\rightarrow$ $<time\_signature><ts\_stave>$ | $<ts\_stave>$
$<time\_signature>$ $\rightarrow$ C | $\frac{4}{4}$ | $\frac{3}{4}$ | $\frac{6}{8}$ | etc.
$<ts\_stave>$ $\rightarrow$ $<measure><barline><ts\_stave>$ | $<measure><barline>$
$<barline>$ $\rightarrow$ $<beg\_barline>$ | $<simple\_barline>$ | $<double\_barline>$ |*etc.*
$<measure>$ $\rightarrow$ $<change\_of\_k\_sign.><ks\_measure>$ | $<ks\_measure>$
$<change\_of\_k\_sign.>$ $\rightarrow$ $<key\_signature>$
$<ks\_measure>$ $\rightarrow$ $<change\_of\_t\_sign.><ts\_measure>$ | $<ts\_measure>$
$<change\_of\_t\_sign.>$ $\rightarrow$ $<time\_signature>$
$<ts\_measure>$ $\rightarrow$ $<vertical\ event><ts\_measure>$ | $<vertical\ event>$
$<vertical\ event>$ $\rightarrow$ $<stem><vertical\ event>$ | $<stem>$

$<stem>$ $\rightarrow$ $<beams><note\_stem>$|$<flags><note\_stem>$|$<note\_stem>$|$<rest>$

$<beams>$ $\rightarrow$ left-beam $<beams>$ | right-beam $<beams>$ | left-beam | right-beam
$<flags>$ $\rightarrow$ flag $<flags>$ | flag

$<note\_stem>$ $\rightarrow$ note-head $<note\_stem>$ | note-head stem

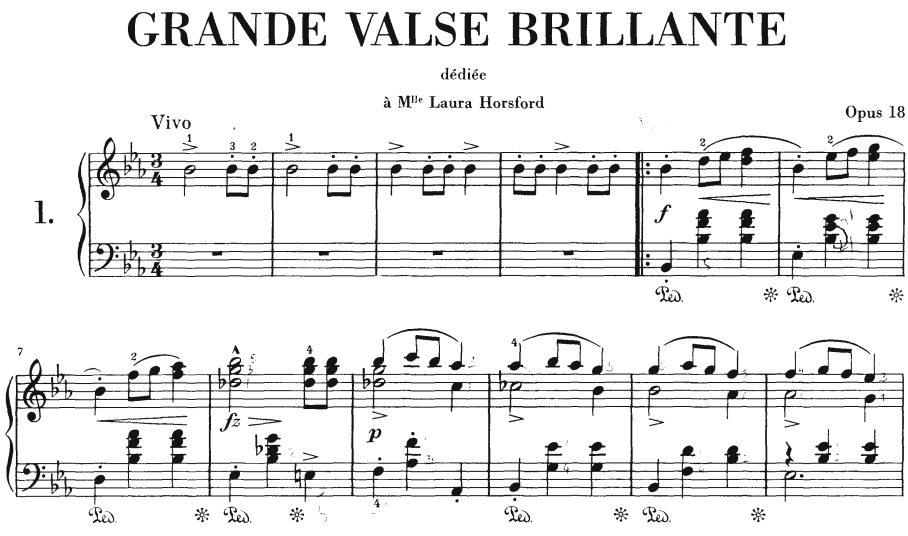


Fig. 1. An example of music notation: the first 12 measures of Chopin's Grand Valse Brillante in E-flat Major, Op. 18

It is worth to notice that the grammar allows to describe two dimensional constructions of music notation. For instance, the following productions allow generating chords of notes, which are placed evenly vertically.

$$<\textit{vertical event}> \rightarrow <\textit{stem}><\textit{vertical event}> \mid <\textit{stem}>$$

A part of derivation tree (Figure 2) of the printed music notation presented in Figure 1. This part shows derivation of the first measure of the music. Further parts of the tree could be derived analogously.

2.2 Braille Music

Printed music notation is a two dimensional language. Valuation of many symbols of this language depends on their horizontal and vertical placement. Moreover, mutual placement of symbols often is as important as symbols themselves. This feature, easily eyesight recognized, raises problems for blind people. This is why a direct conversion to some two dimensional language is not applicable. In past decades a music description language for blind people, so called Braille music, has been invented. Braille music is a linear language. Due to this feature, the second dimension of printed music notation is usually represented as repeated vertical layers. Both partitions to layers and layers' contents are rule governed. However, such rules are based on deep implicit knowledge. Moreover, these rules allow for diversity of descriptions of the same music items. These features must - of course - be considered in construction of a grammar generating Braille music.

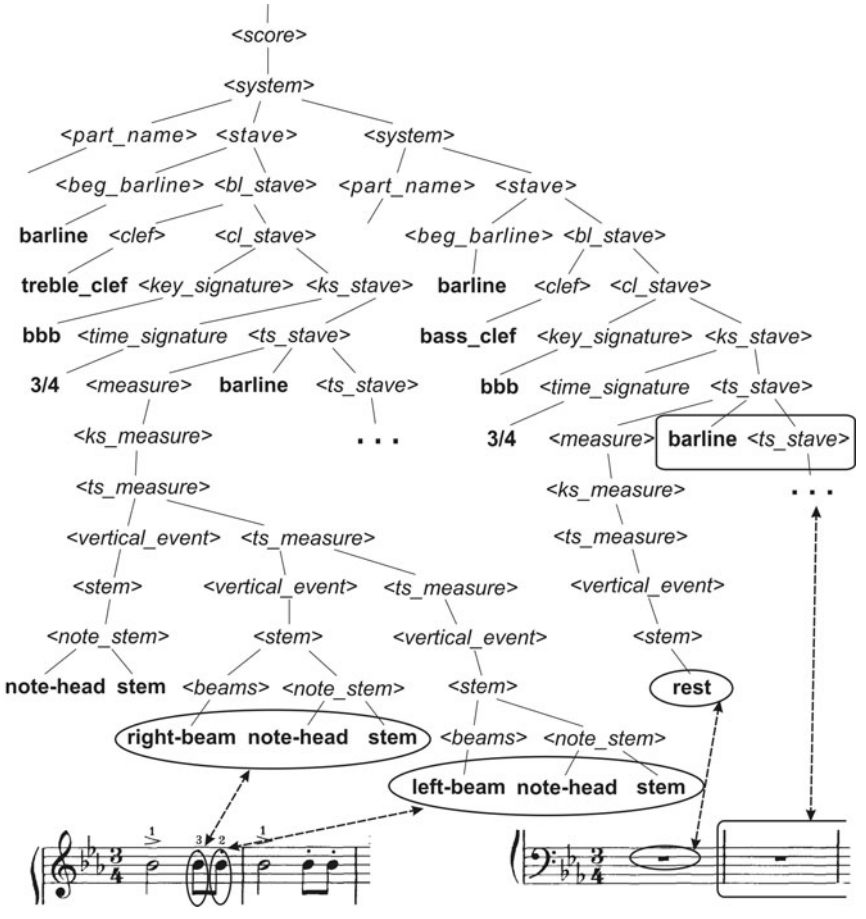


Fig. 2. An excerpt of the derivation tree of printed music notation: the first measure of Chopin's Waltz Op. 18

Linearity of Braille music language makes construction of he grammar more difficult in the sense of bigger number of variants and implied bigger number of productions of the grammar. However, linearity of a grammar generating Braille music makes its construction less sophisticated than construction of a grammar for printed music notation.

In Figure 3 a part of a derivation tree of Braille music is shown. The tree describes the piece of music presented in Figure 1. This part shows derivation of some elements of the first measure of the music. Further parts of the tree could be derived analogously. The grammar generating Braille music is not given in this paper.

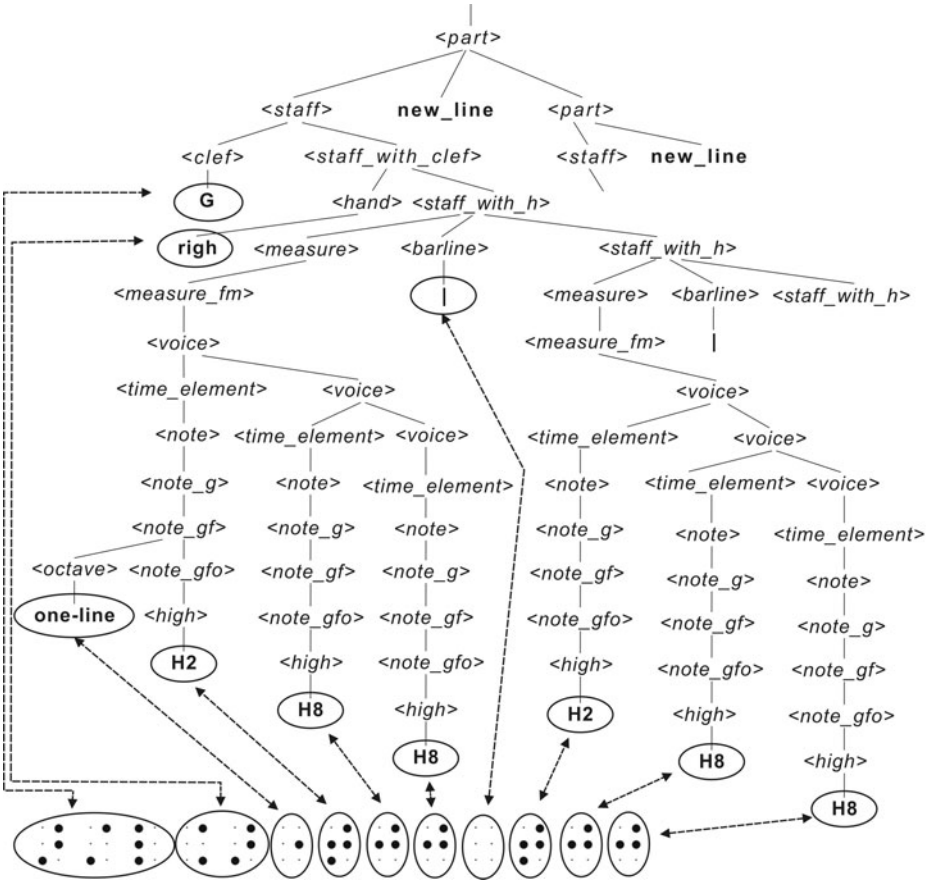


Fig. 3. An excerpt of the derivation tree of Braille music: elements of the first two measures of Chopin's Waltz Op. 18

3 Semantics

Both languages of natural communication discussed in Section 2 describe the same space of real objects. Hearing sensation, i.e. sounds/notes played at a musical instrument are elements of this space, elements described by printed music notation and Braille music. In this paper we assume that notes are represented by a triple $(beginningtime, duration, pitch)$. The tiny triple of properties can be easily attached by adding volume, articulation features etc.

Consequently, the space of language constructions is immersed in the space of sounds. The immersion gives values of real world to language constructions. The immersion defines meaning of language constructions, defines semantics. In the consecutive sections we show that the immersion can be a mapping as well as many-to-many relation.

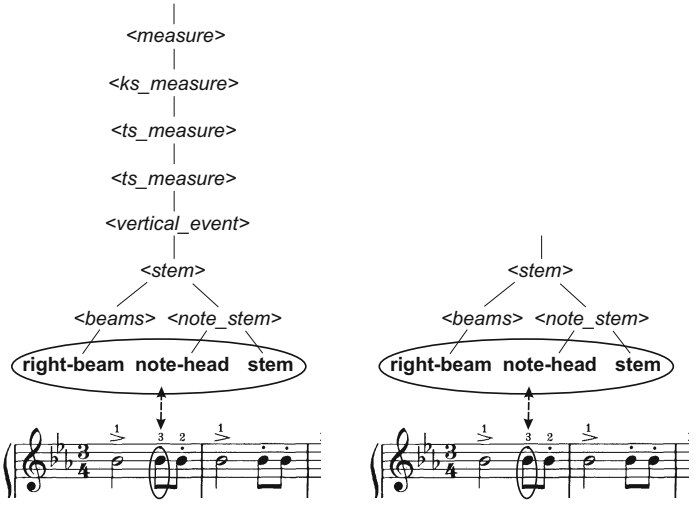


Fig. 4. Printed music notation: samples of lexicon elements

3.1 Semantics of Printed Music Notation

By language constructions we understand a part of music notation with its derivation subtree attached. The space of language constructions (also named statements) is called *lexicon*.

The whole piece of music and its derivation tree is a trivial example of statements, is a trivial element of the lexicon. Figure 4 provides another examples of lexicon elements. Figure 4 explains that the same fragment of music notation may have many derivation subtrees, i.e. may create many elements of the lexicon. This aspect of syntactical structuring, though important for such structural operations on the space of music information like Select and Find/Replace, is out of the scope of this paper.

Formally, the valuation mapping V describes semantics of printed music notation:

$$V : L \rightarrow S$$

where: L is the lexicon of a printed music notation, S is the space of sounds/notes. The mapping V assigns objects of real world S to statements stored in the lexicon L.

The valuation mapping V is not a bijection, so then an inverse mapping cannot be uniquely constructed. Ambiguity of an inverse of the valuation mapping is raised by ambiguity of the language of music notation.

On the other hand, elements of the lexicon L indicated in Figure 4 are the same for more than one eighth note of the notation. For instance, isolated notes of the same duration and pitch are indistinguishable with some lexicon elements. In Figure 2 these elements define the eighth note in the first measure. And, for instance, the same lexicon elements will be utilized for the eighth note in the

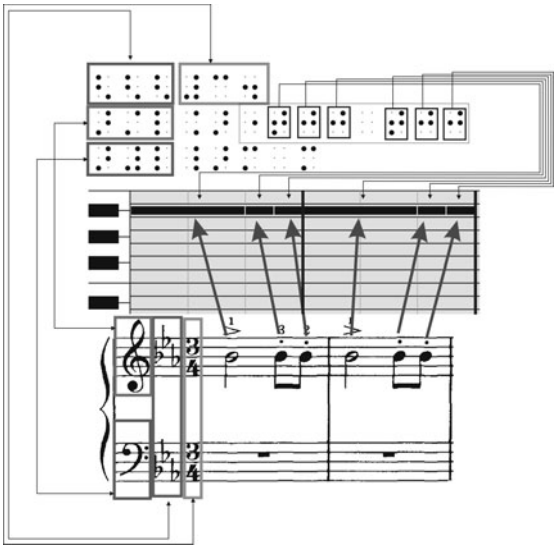


Fig. 5. Valuation function: mappings from printed music notation and Braille music into the space of sounds. Derivation trees are not shown. The played notes are displayed as strips on the scale of piano keyboard.

second measure. Isolated notes of the same duration and pitch are distinguishable with derivation subtree with its own root in the root of the whole derivation tree. This is another interesting aspect stemming out of the space of this paper.

3.2 Semantics of Braille Music

In case of Braille usage lexicon L contains Braille cells and theirs combinations. Braille mapping is relation many-to-many, which means that one Braille cell can describe one or more music notation symbol, and one music notation symbol can be described by one or more Braille cells.

Assume that B is the set of Braille cells, which has a cardinality $2^6 = 64$. Then Braille music tuples (B_C) create the space:

$$B_C = \bigcup_{k=1}^{n} \underbrace{B \times B \times \ldots \times B}_{k \ \ times}$$

where value of n doesn't exceed 10, in practice n can be limited to 5, c.f. [5]).

Tuples of B_C correspond to elements of printed music notation. This correspondence is weakly context dependent, so then it would be comparably easy to construct an almost one-to-one mapping between both languages. However, in practice, many Braille tuples are contracted by dropping Braille cells, which could be context-derived. Such a simplification makes a Braille music easy for reading (by blind people), but heavily ambiguous and context dependent. For instance, a note is described by a triple including a cell defining pitch within an octave and a

duration within one of two intervals. This cell should be preceded by octave indicator cell and duration modifier cell, but both such cells are dropped.

Consequently, semantics of Braille music is defined by the formula:

$$W \in B \times S$$

where: B is the lexicon of a Braille music, S is the space of sounds/notes. The relation W is clearly many-to-many one. It ties objects of real world S with statements stored in the lexicon L.

A glimpse on semantics of Braille music (and of printed music notation) is given in Figure 5. It is shown that some abstract semantic entities (clefs, key and time signatures) are not represented in the sounds/notes space S. Therefore, a space of such intangible items should supplement the space S.

4 Conversions

Creating digital libraries of wide range of documents is an important task of computing equipment. This task often involves technologies recognizing structure of acquired information. In case of music information we mention a paper-to-digital information transfer. Such a transfer requires recognizing printed music notation and storing acquired information structures in a digital form. The OMR (Optical Music Recognition) technology provide tools for conversion of sheet music to a digital form. In this study we discuss further processing of music notation, i.e. conversion of a digital representation of sheet music to a Braille music. Such a conversion could be done by a musician knowing Braille music. However, it is desirable to employ an automaton for such a boring task. Automation of sheet music to Braille music conversion would be easily automated assuming nonambiguity of both languages and a one-to-one mapping between them. Since, as it is pointed out in previous sections, none of these conditions is satisfied, then it is necessary to employ much more sophisticated methods for the conversion(s).

Formally, conversion from a piece of printed music to a piece of Braille music is matter of construction of a mapping C:

$$C : L \to B$$

where: L is the lexicon of a printed music piece nd B is the Braille music lexicon. However, automatic construction of such a mapping is extremely hard, if possible at all, c.f. [2]. A flawless solution has been reported in [6]. The conversion was done as mappings

$$L \xrightarrow{V} S \xrightarrow{W^{-1}} B$$

and

$$B \xrightarrow{W} S \xrightarrow{V^{-1}} L$$

where V^{-1} and W^{-1} are backward (*some* inverse) operations to mapping V and relation W. Construction of inverse of mapping function was the main issue of this conversions. It is achievable e.g. by limiting the space S, by simplifying

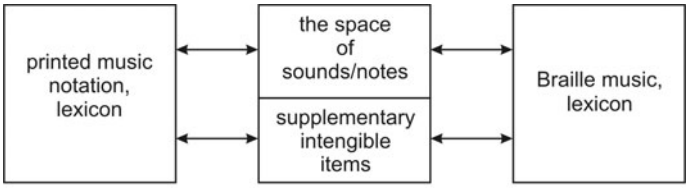


Fig. 6. The structure of conversions between printed music notation and Braille music

lexicon, by using deep, language specific analysis or by analyzing wider part of entirety at once. The solution has been accepted by authorities in Braille music and has been proved by round conversions: $V \circ W^{-1} \circ W \circ V^{-1}$ and $W \circ V^{-1} \circ V \circ W^{-1}$, c.f. [6]. Finally, the space S of sounds/notes is supplemented by abstract items, as mentioned in the recent section. The final structure of conversions is shown in Figure 6.

5 Conclusions

Conversions between different languages of music description, namely: printed music notation and Braille music, are studied in this paper. Conversions between both languages, regarded as languages of natural communication, are difficult to be automated. In fact, structural operations on these languages, including conversions, still raise technological challenge. Successful practical tasks prove that conversions (and other structural operations) should be regarded in broader frames of paradigms of granular structuring and granular computing and should involve syntactic and semantic data processing.

References

1. Bargiela, A., Pedrycz, W.: Granular mappings. IEEE Transactions on Systems, Man, and Cybernetics-part A 35(2), 292–297 (2005)
2. Grant no N R02 0019 06/2009, Breaking accessibility barriers in information society. Braille Score - a computer music processing for blind people, Institute for System Research, Polish Academy of Sciences, report, Warsaw (2011)
3. Homenda, W.: Automatic data understanding: A necessity of intelligent communication. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS (LNAI), vol. 6114, pp. 476–483. Springer, Heidelberg (2010)
4. Homenda, W.: Towards automation of data understanding: integration of syntax and semantics into granular structures of data. In: Fourth International Conference on Modeling Decisions for Artificial Intelligence, MDAI 2007, Kitakyushu, Fukuoka, Japan, August 16-18 (2007)
5. Krolick, B.: How to Read Braille Music, 2nd edn. Opus Technologies (1998)
6. Sitarek, T.: Conversions between MusicXML and Braille Music, B.Sc. Thesis, Warsaw University of Technology, Warsaw (2011) (in Polish)

All That Jazz in the Random Forest

Elżbieta Kubera[1], Miron B. Kursa[2], Witold R. Rudnicki[2], Radosław Rudnicki[3], and Alicja A. Wieczorkowska[4]

[1] University of Life Sciences in Lublin, Akademicka 13, 20-950 Lublin, Poland
elzbieta.kubera@up.lublin.pl
[2] Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, Pawińskiego 5A, 02-106 Warsaw, Poland
{M.Kursa,W.Rudnicki}@icm.edu.pl
[3] The University of York, Department of Music, Heslington, York, YO10 5DD, UK
radek.rudnicki@york.ac.uk
[4] Polish-Japanese Institute of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland
alicja@poljap.edu.pl

Abstract. In this paper, we address the problem of automatic identification of instruments in audio records, in a frame-by-frame manner. Random forests have been chosen as a classifier. Training data represent sounds of selected instruments which originate from three commonly used repositories, namely McGill University Master Samples, The University of IOWA Musical Instrument Samples, and RWC, as well as from recordings by one of the authors. Testing data represent audio records especially prepared for research purposes, and then carefully labeled (annotated). The experiments on identification of instruments on frame-by-frame basis and the obtained results are presented and discussed in the paper.

Keywords: Music information retrieval, Sound recognition, Random forests.

1 Introduction

Music information retrieval is a broad field, addressing various needs of potential users of audio, or audio-visual data [18]. This may include finding a title and a performer of a given excerpt, represented as an audio file taken from a CD, or hummed by the user. Systems allowing such queries already exist, see for example [19] or [25]; style or genre of the given piece of music can also be classified using various systems [24]. A more advanced (or rather more musically educated) user may also want to extract the score, to play it with colleagues, if the score is not very complicated, or maybe is simplified, in order to make such semi-amateur playing feasible. Extraction of pitch, so-called pitch tracking, is already performed in query-by-humming systems. Multi-pitch tracking is a more challenging issue, and some of musical sounds have no definite pitch. Still, if pitches are extracted from a piece of music, labeling them with timbre can aid extracting the score, because then the identified notes can be assigned to the

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 543–553, 2011.

voices (instruments). Our goal is to identify musical instruments in polyphonic environment, when several instruments can be playing at the same time.

Recognition of a much simpler case, i.e. a single instrument playing a single isolated sound, has already been investigated by various researchers. Results vary, depending on the number of instruments taken into account, a feature vector used, and a classifier applied, as well as the validation method utilized. Accuracy for four instruments can reach 100%, and generally decreases with increasing number of instruments, even below 40% when the number of instruments approaches thirty. Also, since audio data are parameterized before applying classifiers, the feature vector representing the analyzed sounds also strongly influences the obtained results.

The feature set can be based on the spectrum obtained from the sound analysis, or the time-domain based description of the sound amplitude or its spectrum. Fourier transform is commonly used here, but other analyzes can be applied as well, e.g. wavelet transform. Although there is no standard set of parameters used in the research on musical instrument recognition, low-level audio descriptors from the MPEG-7 standard of multimedia content description [8] are quite often used. Also, MFCC (Mel-Frequency Cepstral Coefficients), originating from speech recognition, are sometimes applied in music information retrieval [4], and the use of cepstral coefficients includes recognition of musical instruments [2]. Since we have already performed similar research, we decided to use MPEG-7 based sound parameters, as well as additional ones [13].

The accuracy of identification of timbre (i.e. of the instrument) not only depends on the parameterization applied, but also on the classifier used. Many classification methods have already been applied to the task of musical instrument identification, including k-nearest neighbors (k-NN), artificial neural networks (ANN), rough-set based classifiers, support vector machines (SVM), Gaussian mixture models (GMM), decision trees and random forests, etc. The reviews of the methods used in this research is presented in [6], [7] and [9]. The accuracy of such recognition is still far from being perfect, if a number of instruments to recognize is more than, say, a dozen. Still, even simple algorithms, like k-NN, may yield good results, but they were basically applied in the research on isolated monophonic sounds, and when tried on duets, they are prone to errors [17]. This is because in case of polyphonic and polytimbral (i.e. representing plural instruments) recordings, with more than one sound present at the same time, the recognition of instruments becomes much more challenging. Therefore, the use of more sophisticated algorithms seems to be more appropriate. For example, in [10] ANN yielded over 80% accuracy for several 4-instrument sets; GMM yielded about 60% accuracy for duets from 5-instrument set [3].

In our previous research [13], we obtained about 80% accuracy using random forests to classify musical instrument sounds in polytimbral audio data, for 2–5 simultaneous sounds from 14-instrument set. We also used SVM which gained popularity in recent years [12], [27]; still, random forests outperformed SVM by an order of magnitude in our research. This is why we decided to continue experiments using this technique. Our previous research described identification of

complete sound events, i.e. the beginning and the end of each sound (being a note or a chord) must have been identified. In this paper, we cope with recognition of instruments in recordings of frame-by-frame basis, so no prior information on the borders of notes is required. These recordings were especially prepared for the purpose of this research, including the recording and the ground-truth labeling of audio data.

2 Feature Vector

The audio data we use represent simultaneously played sounds of plural pitches. Therefore, the feature set we applied in this research is not based on pitch, thus avoiding multi-pitch identification and possible errors regarding labeling particular sounds with the appropriate pitches. The features describe properties of an audio frame of 40 ms length, subjected to Fourier transform, windowed with Hamming window, and hop size 10 ms. Additionally, for each basic feature, the difference between the feature value calculated for a 30 ms sub-frame of the given 40 ms frame (starting from the beginning of this frame) and next 30 ms sub-frame (starting with 10 ms offset) was also added to the feature set. Most of these features come from MPEG-7 low-level audio features [8]. Therefore, the following features constitute our feature vector:

- *AudioSpectrumFlatness*, $flat_1, \ldots, flat_{25}$ - multidimensional parameter describing the flatness property of the power spectrum (obtained through the Fourier transform) within a frequency bin for selected bins; 25 out of 32 frequency bands were used for a given frame;
- *AudioSpectrumCentroid* - power weighted average of the frequency bins in the power spectrum; coefficients are scaled to an octave scale anchored at 1 kHz [8];
- *AudioSpectrumSpread* - a RMS (root mean square) value of the deviation of the Log frequency power spectrum with respect to *AudioSpectrumCentroid* for the frame [8];
- *Energy* - energy (in logarithmic scale) of the spectrum of the parameterized sound;
- $MFCC$ - vector of 13 Mel frequency cepstral coefficients. The cepstrum was calculated as logarithm of the magnitude of the spectral coefficients, and then transformed to the mel scale, used instead the Hz scale, in order to better reflect properties of the human perception of frequency. Twenty-four mel filters were applied, and the obtained results were transformed to twelve coefficients. The thirteenth coefficient is the 0-order coefficient of MFCC, corresponding to the logarithm of the energy [11], [20];
- *ZeroCrossingRate*; zero-crossing is a point where the sign of time-domain representation of sound wave changes;
- $RollOff$ - the frequency below which an experimentally chosen percentage equal to 85% of the accumulated magnitudes of the spectrum is concentrated. It is a measure of spectral shape, used in speech recognition to distinguish between voiced and unvoiced speech;

- $NonMPEG7 - AudioSpectrumCentroid$ - a differently calculated version - in linear scale;
- $NonMPEG7 - AudioSpectrumSpread$ - a different version; the deviation is calculated in linear scale, wrt. $NonMPEG7 - AudioSpectrumCentroid$;
- changes (measured as differences) of the above features for 30 ms subframes,
- $Flux$ - the sum of squared differences between the magnitudes of the DFT points calculated for a 30 ms sub-frame of the given 40 ms frame (starting from the beginning of this frame) and the next 30 ms sub-frame (starting with 10 ms offset); this feature by definition describes changes of magnitude spectrum, thus it cannot be calculated in a static version.

The features from the MPEG-7 set can be considered a standard in audio indexing tasks. Since the random forest classifier handles the redundant features without degrading the performance, hence we did not carry any feature selection for this particular problem.

Our previous research [13], [14] was performed for complete sound events (chords), and parameters derived from the entire sound were then included in the feature set, consisting of several features describing changes in time and static features presented here. We obtained good results of automatic identification of a predominant instrument in sound mixes, so these features turned out to be useful in a practical classification task. The importance of these features was tested through the *Boruta* feature selection algorithm [15], [16]. The experiments were conducted for various levels of additional sounds accompanying the target one; the higher level of added sounds, the more attributes were indicated as important. In case of the data set representing a combined set of mixes for various levels of added sounds, all parameters turned were indicated by *Boruta* as important. Moreover, such parameters, for instance $LogAttackTime$, $TemporalCentroid$, were indicated as very significant by the *Boruta* feature selection algorithm [15], [16], and most often selected in our Random Forest classification process. Since now we are going to identify small portions of audio data (frames), representing only a fraction of a sound, we cannot use parameters describing the entire sound, so they are absent in our feature vector. This makes the task that we put our algorithms to cope with even more difficult.

3 Audio Data

The purpose of our experiments was to identify musical instruments playing in a given piece of music. Therefore, we needed recordings labeled with information on every instrument playing in the piece, namely, where exactly it starts and ends playing a note or a sequence of notes (without gaps). Also, in order to train classifiers to identify particular instruments, we needed recordings of these instruments. Our research was performed on 1-channel 16-bit/44.1 kHz data; in case of stereo recordings, we analyzed the mix (i.e. the average) of both channels.

For the training purposes, we used three repositories of single, isolated sounds of musical instruments, namely McGill University Master Samples [21], The University of IOWA Musical Instrument Samples [26], and RWC Musical Instrument

Sound Database [5]. Clarinet, trombone, and trumpet sounds were taken from these repositories. Additionally, we used sousaphone sounds, recorded by one of the authors, R. Rudnicki, since no sousaphone were available in the above mentioned repositories. Training data were in 16 bit/44.1 kHz format, mono in case of RWC data and sousaphone, and stereo for the rest of the data. In order to prepare our classifier to work on different instrument sets, we added other instruments that can be encountered in similar jazz recordings, namely, double bass, piano, tuba, saxophone, and harmonica.

The testing data originate from jazz band recordings, also prepared by R. Rudnicki [22]. The following pieces were investigated:

- Mandeville by Paul Motian,
- Washington Post March by John Philip Sousa, arranged by Matthew Postle,
- Stars and Stripes Forever by John Philip Sousa, semi-arranged by Matthew Postle - Movement no. 2 and Movement no. 3.

These pieces were first recorded with one microphone (mono) per instrument/ track (clarinet, sousaphone, trombone and trumpet) in 32 bit/48 kHz format, and then rendered to stereo mix 16 bit/44.1 kHz using pan effect with creating stereo track for the final mix and for each original track. We had access to these tracks of each instrument (with cross-talks from neighboring instruments, however) contributing to the final mix. These tracks were used as a basis for segmentation, i.e. creation of ground-truth data.

4 Random Forests

Random Forest (RF) is a classifier consisting of a set of weak, weakly correlated and non-biased decision trees. It has been shown that RF perform quite well and often outperform other methods on a diverse set of classification problems [1].

RF is constructed using a procedure that minimizes bias and correlations between individual trees. Each tree is built using different N-element bootstrap sample of the training N-element set. Since the elements of the sample are drawn with replacement from the original set, roughly 1/3 (called OOB, out-of-bag) of the training data are not used in the bootstrap sample for any given tree.

For a P-element feature vector, p attributes (features) are randomly selected of at each stage of tree building, i.e. for each node of any particular tree in RF ($p \ll P$, often $p = \sqrt{P}$). The best split on these p attributes is used to split the data in the node. The best split is determined as minimizing the Gini impurity criterion, measuring how often an element would be incorrectly labeled if randomly labeled according to the distribution of labels in the subset.

Each tree is grown to the largest extent possible, without pruning. By repeating this randomized procedure M times, a collection of M trees is obtained, constituting a random forest. Classification of an object is done by simple voting of all trees.

4.1 Methodology of RF Training in the Experiments

When preparing training data, each audio file representing a single isolated sound of an instrument is normalized first to RMS equal to one. Next, silence is removed: a smoothed version of amplitude is calculated starting from the beginning of the file, as moving average of 5 consequent amplitude values, and when this value increases by more than a threshold (experimentally set to 0.0001), this point is considered to be the end of the initial silence. Similarly, the ending silence is removed. Then we perform parameterization, and train RF to identify each instrument – even when accompanied by other sound. Therefore, we perform training on 40 ms frames of instrument sounds, mixing from 1 to 4 randomly chosen instruments with random weights and then normalized again to RMS=1. The battery of one-instrument sensitive random forest classifiers is then trained: 3,000 mixes containing any sound of a given instrument are fed as positive examples, and 3,000 mixes containing no sound of this instrument are fed as negative examples. For N instruments we need N binary classifiers – N random forests, each one trained to identify 1 instrument. Quality test of the forest is performed on 100,000 mixes (prepared the same way as the training data), and then RF is ready to be applied to frame-by-frame recognition of instruments in recordings like jazz band mentioned in Section 3.

According to our best knowledge, there is no overfitting problem when RFs are used, and this is actually one of the main advantages of RFs. Adding more trees to the forest never degrades the performance of the classifier. That is the original claim of the authors, which has been proved mathematically.

The claims about overfitting that can be encountered are based on misunderstanding of the results obtained for regression tasks [23]. The described effect of decreasing accuracy, when the number of splits is increased, has nothing to do with actual overfitting, which can happen when increased model complexity increases the fit to the training set, but decreases the fit to the test set. In the case of RF fit both to training set and test set is degraded when the number of splits is not optimal. One should note, however, that this effect was described for regression tasks and not for classification, and only for the trees that were allowed to grow to the full extent. This was rectified by introduction of the minimal node size that is split in the regression tasks. Here we deal with a classification task, and we are not aware of any paper showing the problem of overfitting for the RF classification.

5 Experiments and Results

After training of the RF classifier on sounds of the 9 instruments mentioned in Section 3 (see Table 1), we performed identification of instruments playing in recordings where only 4 instruments were present. Ground-truth data were prepared through careful manual labeling by one of the authors, based on audio data for each instrument track separately. Precise labeling of the beginning and end of each note is not an easy task; onsets can be relatively easily found, but finding the end can be difficult because the sound level diminishes gradually,

Table 1. The results of quality test of RF training, for mixes created with weights w exceeding the indicated levels for the target instrument

	$w = 0.5$	$w = 0.4$	$w = 0.3$	$w = 0.2$	$w = 0.1$	$w = 0.0$
Instrument	Accuracy [%]					
clarinet	80.3	76.1	72.1	69.5	67.4	66.6
double bass	84.1	81.3	79.0	76.8	74.3	72.0
harmonica	79.4	82.7	87.9	92.7	96.0	96.6
piano	80.6	79.1	77.1	75.5	72.6	69.6
saxophone	83.5	80.2	78.2	75.9	74.5	71.4
sousaphone	84.6	81.3	79.3	78.2	76.4	74.2
trombone	85.7	83.4	82.8	81.9	80.5	77.5
trumpet	85.3	83.5	83.1	83.2	81.7	78.8
tuba	87.0	87.4	89.3	90.9	90.1	87.2

merging with ambient sounds and "silence" that is not a flat line in the audio track. Therefore, there are segments that were labeled as neither presence or absence of a given instrument.

In Table 2 we present the results of identification of instruments on frame-by-frame basis, for frames clearly labeled as the ones where a given instrument is present or absent. In order to take into account loudness of each instrument, the results were weighted by RMS values measured for original tracks of each instrument independently. As we can see, precision of the RF instrument identification is very high, but recall is rather low; which means that if the instrument is identified, it usually is identified correctly, but in many cases the instrument may not be recognized. Therefore, it would be interesting to see which frames pose difficulties to RF classification. It is illustrated in Figure 1. Annotation (Ann) shows ground-truth data, weighted by RMS for a given instrument track; prediction (Pred) shows prediction obtained as confidence level yielded by RF, weighted by RMS of the entire sample (all normalized). Darker levels of gray correspond to higher results; segments not annotated with neither presence nor absence of a given instrument are marked with slanting black lines. As we can see, generally identification of sound events by RF follows annotation, although border areas are problematic. However, we must be aware that borders should not be considered to be crisp, because annotation could cover bigger or smaller intervals, depending on the levels of the investigated sounds. Also, single errors can be removed by post-processing, and labeling single outliers according to the neighboring frame labels.

As we mentioned before, we performed training on 8 instruments, and one can be interested to see which instruments tend to be confused. This is illustrated in Figure 2, showing 2D projection of instrument misclassification rate (i.e. of the 8x8 contingency table). The closer the instruments, the more often they are confused. As we can see, there is no close proximity between any of these instruments, so there are no strong tendencies to confuse any particular of these instruments with each other.

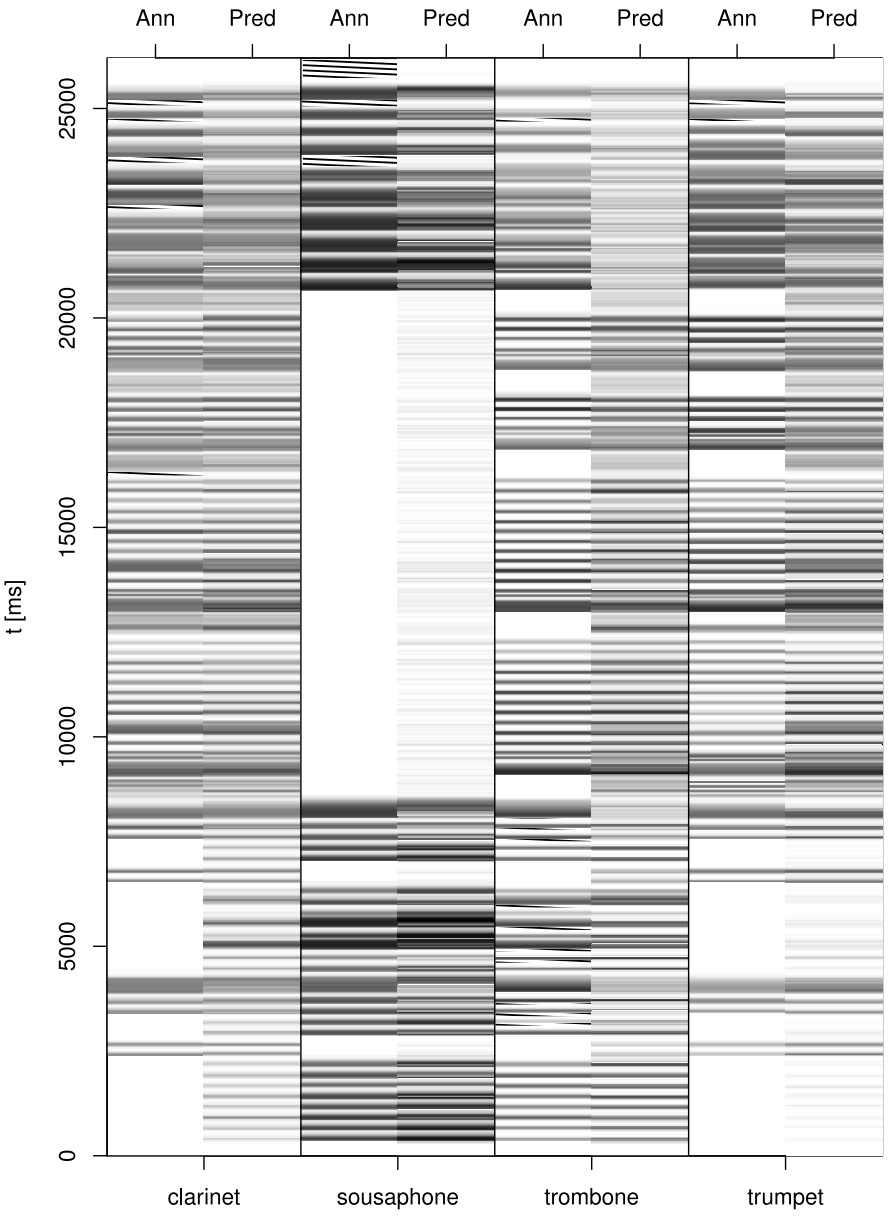


Fig. 1. Results of RF recognition of instruments in "Stars and Stripes", 3rd movement. The left hand side of each column (Ann) indicates annotated areas, weighted by the sound intensity, while the hand right side (Pred) shows the intensity predicted by the RF classifier. Darker color indicates higher intensity; striped areas were not annotated with neither presence nor absence of a given instrument

Table 2. Classification results for jazz band recordings, weighted by RMS measured for each instrument track

Mandeville				
	clarinet	sousaphone	trombone	trumpet
precision	97.10%	99.64%	98.26%	99.53%
recall	53.88%	65.78%	47.51%	46.66%
Washington Post				
	clarinet	sousaphone	trombone	trumpet
precision	93.06%	97.51%	84.63%	98.66%
recall	47.25%	59.19%	42.76%	60.89%
Stars and Stripes, Movement no. 2				
	clarinet	sousaphone	trombone	trumpet
precision	79.65%	99.65%	99.34%	99.97%
recall	36.33%	62.06%	28.89%	50.43%
Stars and Stripes, Movement no. 3				
	clarinet	sousaphone	trombone	trumpet
precision	98.92%	99.87%	99.10%	98.44%
recall	45.06%	39.32%	23.03%	60.31%

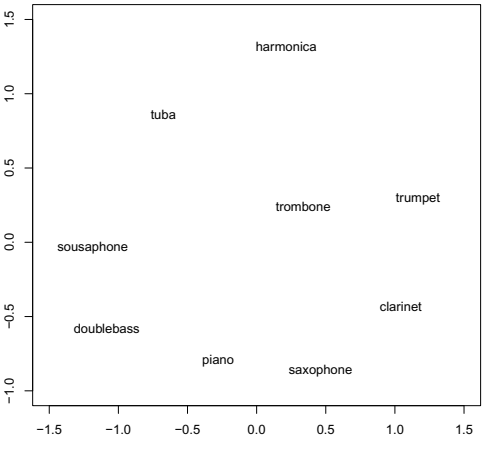


Fig. 2. 2D projection of instrument misclassification rate (i.e. of the 8×8 contingency table). The closer the instruments, the more often they are confused.

6 Summary and Conclusions

Identification of musical instruments in audio recordings is not an easy task. In this paper, we report results of using RF for automatic recognition of instruments in audio data, on frame-by-frame basis. Assessment and presentation of results are also challenging in this case, because even annotation of such recordings is challenging for humans if precise segmentation is needed, and illustrative comparison of ground-truth and the obtained results of automatic classification requires taking into account several dimensions of sound: not only timbre (i.e. instrument), but also loudness, and changes of sound waves in time. The recordings of jazz band we investigated represented 4 instruments, but we trained RF classifier to recognize 8 instruments, so such automatic recognition of instruments can be extended to other jazz recordings. The same methodology can be also applied to other sets of instruments.

The outcomes of the presented experiments show that recall is relatively low, so some sounds are left unrecognized, but precision is high, so if an instrument is identified, this identification is performed with high confidence. Therefore we believe that the presented methodology can be applied to identification of segments played by musical instruments in audio records, especially if post-processing is added to clean the results.

Acknowledgments. The authors would like to express thanks to the musicians who recorded our test audio data: Joseph Murgatroyd (clarinet), Matthew Postle (trumpet), Noah Noutch (trombone) and James M. Lancaster (sousaphone). We are also grateful to Matthew Postle for arrangements of these pieces.

This project was partially supported by the Research Center of PJIIT, supported by the Polish National Committee for Scientific Research (KBN). Computations were performed at ICM, grant G34-5.

References

1. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001), http://www.stat.berkeley.edu/~breiman/RandomForests/cc_papers.htm
2. Brown, J.C.: Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. J. Acoust. Soc. Am. 105, 1933–1941 (1999)
3. Eggink, J., Brown, G.J.: Application of missing feature theory to the recognition of musical instruments in polyphonic audio. In: ISMIR (2003)
4. Foote, J.: An Overview of Audio Information Retrieval. Multimedia Systems 7(1), 2–11 (1999)
5. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC Music Database: Music Genre Database and Musical Instrument Sound Database. In: Proceedings of ISMIR, pp. 229–230 (2003)
6. Herrera, P., Amatriain, X., Batlle, E., Serra, X.: Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In: International Symposium on Music Information Retrieval, ISMIR (2000)

7. Herrera-Boyer, P., Klapuri, A., Davy, M.: Automatic Classification of Pitched Musical Instrument Sounds. In: Klapuri, A., Davy, M. (eds.) Signal Processing Methods for Music Transcription, Springer Science+Business Media LLC (2006)
8. ISO: MPEG-7 Overview, http://www.chiariglione.org/mpeg/
9. Klapuri, A., Davy, M. (eds.): Signal Processing Methods for Music Transcription. Springer, New York (2006)
10. Kostek, B.: Musical Instrument Classification and Duet Analysis Employing Music Information Retrieval Techniques. Proc. IEEE 92(4), 712–729 (2004)
11. Kubera, E.: The role of temporal attributes in identifying instruments in polytimbral music recordings (in Polish). Ph.D. dissertation, Polish-Japanese Institute of Information Technology (2010)
12. Kubera, E.z., Wieczorkowska, A., Raś, Z., Skrzypiec, M.: Recognition of instrument timbres in real polytimbral audio recordings. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6322, pp. 97–110. Springer, Heidelberg (2010)
13. Kursa, M.B., Kubera, E.z., Rudnicki, W.R., Wieczorkowska, A.A.: Random musical bands playing in random forests. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS (LNAI), vol. 6086, pp. 580–589. Springer, Heidelberg (2010)
14. Kursa, M., Rudnicki, W., Wieczorkowska, A., Kubera, E.z., Kubik-Komar, A.: Musical instruments in random forest. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 281–290. Springer, Heidelberg (2009)
15. Kursa, M.B., Jankowski, A., Rudnicki, W.R.: Boruta: A System for Feature Selection. Fundamenta Informaticae 101, 271–285 (2010)
16. Kursa, M.B., Rudnicki, W.R.: Feature Selecion with the Boruta Package. J. Stat. Soft. 36, 1–13 (2010)
17. Livshin, A.A., Rodet, X.: Musical Instrument Identification in Continuous Recordings. In: Proc. DAFX 2004 (2004)
18. Müller, M.: Information retrieval for music and motion. Springer, Heidelberg (2007)
19. MIDOMI, http://www.midomi.com/
20. Niewiadomy, D., Pelikant, A.: Implementation of MFCC vector generation in classification context. J. Applied Computer Science 16(2), 55–65 (2008)
21. Opolko, F., Wapnick, J.: MUMS – McGill University Master Samples. CD's (1987)
22. Rudnicki, R.: Jazz band. Recording and mixing. Arrangements by M. Postle. Clarinet - J. Murgatroyd, trumpet - M. Postle, harmonica, trombone - N. Noutch, sousaphone - J. M. Lancaster (2010)
23. Segal, M.: Machine Learning Benchmarks and Random Forest Regression. Center for Bioinformatics & Molecular Biostatistics, http://repositories.cdlib.org/cbmb/bench_rf_regn/
24. Shen, J., Shepherd, J., Cui, B., Liu, L. (eds.): Intelligent Music Information Systems: Tools and Methodologies. Information Science Reference, Hershey (2008)
25. Sony Ericsson: TrackID, http://www.sonyericsson.com/trackid/
26. The University of IOWA Electronic Music Studios: Musical Instrument Samples, http://theremin.music.uiowa.edu/MIS.html
27. Wieczorkowska, A.A., Kubera, E.: Identification of a dominating instrument in polytimbral same-pitch mixes using SVM classifiers with non-linear kernel. J. Intell. Inf. Syst. 34(3), 275–303 (2010)
28. Zhang, X., Marasek, K., Raś, Z.W.: Maximum Likelihood Study for Sound Pattern Separation and Recognition. In: 2007 International Conference on Multimedia and Ubiquitous Engineering, MUE 2007, pp. 807–812. IEEE, Los Alamitos (2007)

Selection of the Optimal Microelectrode during DBS Surgery in Parkinson's Patients

Konrad Ciecierski[1], Zbigniew W. Raś[2,1], and Andrzej W. Przybyszewski[3]

[1] Warsaw Univ. of Technology, Institute of Comp. Science, 00-655 Warsaw, Poland
[2] Univ. of North Carolina, Dept. of Comp. Science, Charlotte, NC 28223, USA
[3] UMass Medical School, Dept. of Neurology, Worcester, MA 01655, USA
konrad.ciecierski@gmail.com, ras@uncc.edu,
Andrzej.Przybyszewski@umassmed.edu

Abstract. Deep brain stimulation (DBS) of the subthalamic nucleus (STN) is effective treatment of Parkinson disease. Because the STN is small ($9 \times 7 \times 4mm$) and it is not well visible using conventional imaging techniques, multi-microelectrode recordings are used to ensure accurate detection of the STN borders. Commonly used discriminations which microelectrode's signal relates to the activity of the STN are signal quality and neurologist's experience dependent. The purpose of this paper is to determine the STN coordinates in a more objective way. We present analysis of the neurological signals acquired during DBS surgeries. The purpose of our method is to discover which one of the scanning microelectrodes reaches the target area guaranteeing a most successful surgery. Signals acquired from microelectrodes are first filtered. Subsequently the spikes are detected and classified. After that, new signal is reconstructed from spikes. This signal's power is then calculated by means of FFT. Finally cumulative sum of the signal's power is used to choose a proper electrode.

The ultimate goal of our research is to build a decision support system for the DBS surgery. A successful strategy showing which of the recording microelectrodes should be replaced by the DBS electrode is probably the most difficult and challenging.

Keywords: Parkinson's disease, DBS, STN, wavelet, filtering, PCA, FFT, spike detection, spike discrimination, spike clustering.

Introduction

The Parkinson's disease (PD) is a chronic, progressive movement disorder that affects the lives of at least one million patients across the United States and the number of PD patient is constantly increasing as effect of the population. The characteristic motor symptoms of PD, predominantly due to progressive degeneration of nigral dopaminergic neurons, are initially subtle and impact purposeful movement, and are often difficult to diagnose and to differentiate from other age related symptoms. Among it's symptoms there is an impairment of motor skills: tremor, stiffness and slowness of voluntary movements.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 554–564, 2011.

Subthalamic nucleus (STN) deep brain stimulation (DBS) has become the standard treatment for patients with advanced Parkinson's disease (PD) who have intolerable drug-induced side effects or motor complications after the long-term use of dopaminergic drugs. In this surgical procedure microelectrodes are inserted into brain on the track towards estimated from the MRI STN position. When they reach the destination, signal from them is being analyzed and upon the result of this analysis the trajectory of one of them is later used for implantation of the permanent DBS electrode. When the permanent electrode is activated, it disrupts abnormal activity of the STN and the impairment of motor skills to some degree lessens. To minimize the collateral damage to the brain tissue, it is imperative to use as few probing electrodes as possible, and to find the correct trajectory in most precise way.

1 Initial Signal Analysis

1.1 Removal of Low Frequency Components

Recorded signal has to be initially processed before further analysis can begin. Often the signal is contaminated with low frequency components. This low frequencies comes both from biological and non-biological sources. One source in particular is worth mentioning - it's the frequency of power grid 50 Hz in Europe and 60 Hz in US. Below, a raw signal is shown (see Fig. 1) that was actually recorded within patients brain. In this recording one can clearly see that signal has strong component of low frequencies. This low frequencies affects the amplitude of the signal and the same it is very difficult to make any amplitude-based analysis. This is why signal needs to be filtered. All frequencies below 375 Hz and above 3000 Hz were removed and the resulting signal (see Fig. 2) is much more suited for further use.

1.2 Spike Shape Retention

The process of high band filtering absolutely must retain the spikes that were observed in the raw signal. If only the presence of the spikes must be preserved, one

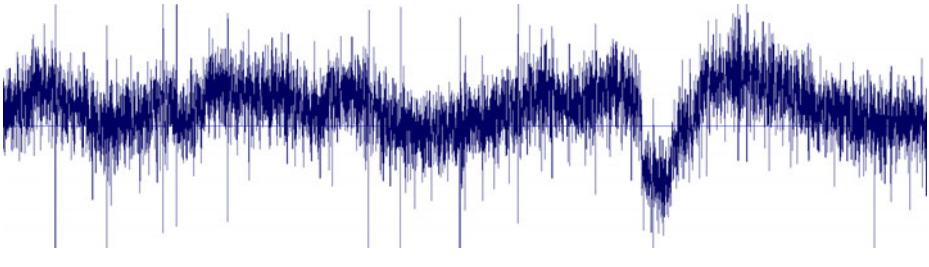


Fig. 1. Intraoperative 1s microelectrode recording from the sunthalamic area. The low frequency oscillations are clearly visible. One can also clearly see 10 spikes having amplitude much larger than the rest of the signal, but there are many other spikes with smaller amplitudes.

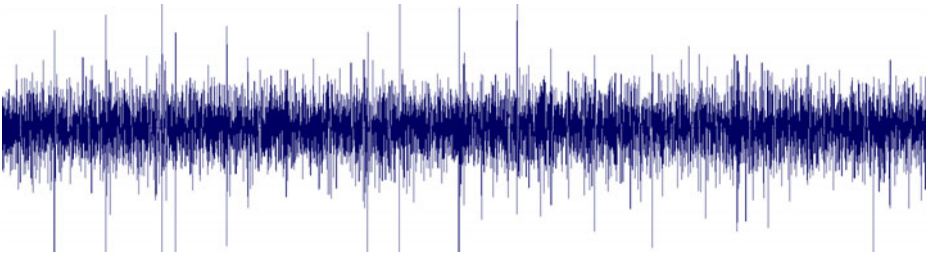
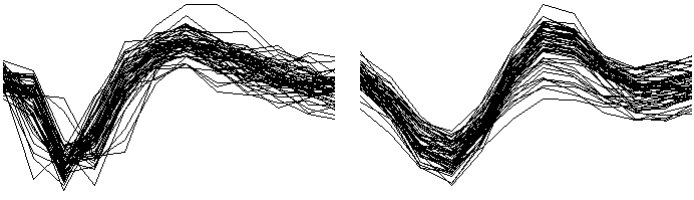


Fig. 2. The same 1s signal as in Fig. 1. but with frequencies below 375 Hz removed. One can still clearly see 10 spikes.



(a) Original shape of unfiltered spikes

(b) Distorted shapes present in FFT filtered signal.

Fig. 3. Comparison of spike shapes in raw (a) and FFT filtered (b) signal

can use FFT filtering. Using FFT is not suitable when not only occurrence but also the shape of the spikes should be preserved. This is because FFT strongly interferes with the shape of the spikes (see Fig. 3). So, this is why Daubechies D4 wavelet filtering was chosen as the filtering method. This idea of wavelet decomposition, filtering and reconstruction of DBS recorded signal has been described in [4]

2 Spike Detection

As it was stated in [2], it is possible to detect spikes occurring in cells within radius of 50 μm from the electrode's recording tip. Still, this small area may contain around 100 neuronal cells. Recorded spikes from such area with one microelectrode may have different widths, shapes and amplitudes. Cells that are close to electrode will be recorded with a higher amplitudes then those being distal. Distance between electrode and soma can also have an influence on the width of the recorded spike [3]. The greater the distance, the wider become recorded spikes from the same cell. All above makes the task of detecting spikes and discriminating them from the noise even more difficult. Two approaches to spike detection were considered:

2.1 Derivative Approach

In this approach spikes detection bases on the slope of their amplitude. If the first derivative is below and then above some given thresholds during some consistent period of time then it is assumed that spike might have occurred. Now, assuming that amplitude over time is represented by $f(t)$, its derivative as $f'(t)$, lower threshold as d_l and upper threshold as d_u, the necessary condition for spike to occur around time t_0 is shown on equation 1. Knowing that $d_l < 0$ and $d_u > 0$, equations 2 and 3 guarantee, that in the (t_b, t_0) interval $f(t)$ is strictly descending and that in the (t_0, t_e) $f(t)$ is strictly ascending. The d_l and d_u controls respectively the pitch of the descent and the ascent. The drawback of this method is that both d_l and d_u values have to be manually specified for each signal being analyzed. Above makes it difficult to apply this method in an automated, unsupervised way.

$$\exists t_b < t_0 < t_e \, (\forall (t_b < t < t_0) f'(t) < 0 \text{ and } \forall (t_0 \leq t < t_e) f'(t) \geq 0) \tag{1}$$

$$\exists (t_b < t_l < t_0) f'(t_l) < d_l \tag{2}$$

$$\exists (t_0 < t_u < t_e) f'(t_u) > d_u \tag{3}$$

It must also be mentioned that spikes with polarity negative to described above do exist and have to be detected in adequate, similar way.

2.2 Amplitude Approach

Assuming that low frequency components have been already filtered out from the signal, one can attempt to detect spikes using amplitude analysis. In [5] it is postulated to use a specific amplitude threshold for spike detection. Assuming that x_k denotes k_{th} sample of input signal, threshold is there given by value V_{thr} (see equation 4) with $\alpha_{thr} \in \langle 4.0,\ 5.0 \rangle$. In this work different α_{thr} are begin used. During spike detection, program checks for spikes with values $5.0, 4.9, \ldots, 4.0$. Spike is assumed to exist when amplitude is lower then $-V_{thr}$ or higher then V_{thr}. If for some recording at given α_{thr} value, at least 200 spikes are found then it is accepted and lower values of α_{thr} are not considered. While further lowering of α_{thr} would probably yield more spikes they would also be more and more noise contaminated. Count of at least 200 spikes is sufficient for further cluster and power analysis. If value of 4.0 is reached and still less then 30 spikes are found, it is assumed that no representative spikes have been found. Advantage of this approach over the previous one is that in this case, the threshold can be calculated automatically. This allows the process of spike detection to be done in unsupervised - automatic way.

$$V_{thr} = \alpha_{thr} \, \sigma_n \quad \text{where} \quad \sigma_n = \frac{1}{0.6745} \, median(|x_1|, \ldots, |x_n|) \tag{4}$$

2.3 Comparison of Approaches

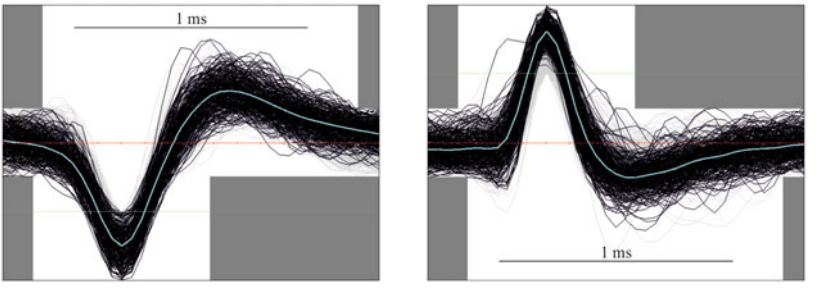
Because of the ability of automatic spike detection, the amplitude approach was chosen. Regardless which approach is selected, some fine tuning is still necessary. This fine tuning is defined as zones of forbidden amplitude and is shown in red (see Fig. 4). In case of $down - up$ spikes (see Fig. 4(a)) the $-V_{thr}$ amplitude is shown as green line. Assuming that amplitude is below $-V_{thr}$ at time t_0 then spike occurs between $t_0 - 0.5\ ms$ and $t_0 + 1.1\ ms$ if fulfilled are conditions (5) $\cdots$ (8). In case of $up - down$ spikes (see Fig. 4(b)) the green line denotes V_{thr} amplitude level. Conditions (5) $\cdots$ (8) must be modified in this case to reflect reversed amplitude.

$$\forall(t_0 - 0.5\ ms < t < t_0 - 0.4\ ms) \quad f(t) > -\frac{V_{thr}}{2} \tag{5}$$

$$\forall(t_0 + 0.4\ ms < t < t_0 + 1.1\ ms) \quad f(t) > -\frac{V_{thr}}{2} \tag{6}$$

$$\forall(t_0 - 0.5\ ms < t < t_0 - 0.3\ ms) \quad f(t) < \frac{V_{thr}}{2} \tag{7}$$

$$\forall(t_0 + 1.0\ ms < t < t_0 + 1.1\ ms) \quad f(t) < \frac{V_{thr}}{2} \tag{8}$$



(a) Down-Up spikes (b) Up-Down spikes

Fig. 4. Forbidden spike amplitude areas: in (a) 264 spikes, in (b) 266 spikes

3 Spike Clustering

As mentioned in section 2, scanning electrode can register spikes coming from about 100 neurons. Not all of them are of the same cell type. Different neurons types/classes have different spike shapes. While it seams that shape alone is not sufficient to determine location of the electrode in patient's brain, it still unsubtly carries information that can be used in further analysis.

3.1 Clustering Using PCA over Spike Amplitude

Archer et. al. in [5] use Principal Component Analysis to obtain principal component vectors and then use mean of the first few to obtain dominant spike shape. Here a modified PCA approach has been applied. Knowing that all spikes from given recording are 1.6 ms wide (see section 2.3) one knows that they are described using the same number of samples. It is so possible to build matrix containing all detected spikes with each row containing single spike. After the PCA is preformed the first ten most important principal component vectors are clustered using $k - means$ method. Resulting cluster gives good spike shape discrimination.

3.2 Clustering Using PCA over Wavelet Decomposition

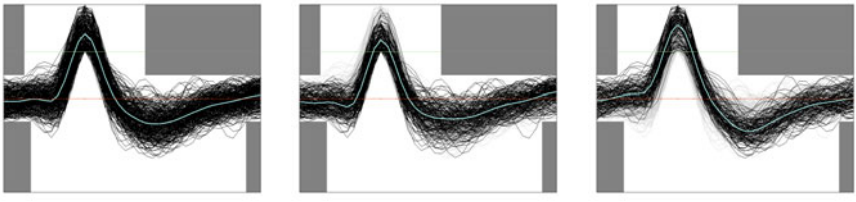
Wavelet decomposition transforms signals recorded in time domain into the frequency domain. It allows one to see the frequency components of given signal. The higher the frequency, the lower becomes the amplitude of the transformed signal. From that observation it becomes obvious that to the shape of the spike, lower frequency components contributes the most and that high frequencies represents summary noise coming from neighbor neurons (see section 2). Comparing the wavelet transforms of different spikes also shows that the lower frequencies are most differentiating. Because of that, to enhance the effectiveness of the clustering, a subset of wavelet transform coefficients is used as the input to the PCA. Following approach used in [6], each spike is transformed using the $4level Haar$, wavelet transform. Matrix containing row by row wavelet transformations of all spikes is then constructed. It is obvious that not all columns are needed for clustering. Some of them (esp. those related to higher frequencies) are redundand. Proper columns are chosen using modified Kolomogorov-Smirnov test (see [6]). The outcome of the PCA is used to obtain clusters in the same way as in (3.1).

3.3 Clustering Summary

Both types of clustering produce good spike shape discrimination. In the (3.1) approach all spike data are being used as an input into PCA. This sometimes produces additional clusters. This clusters represent shapes that are similar to shapes yielded by other clusters and are different mainly in higher frequencies. Table 1 shows comparison of clustering results run on 96 recordings. For each clustering type it is shown how many recordings produced given number of different clusters/shapes. In both clustering approaches most recordings produced only 2 different shapes (73 recordings for amplitude based, and 62 recordings for wavelet based). Only in 9 and 15 respectfully recordings single shape class was detected. Fig. 5 shows example of spike discrimination. In processed recording 312 spikes were found (Fig. 5(a)). Spikes were subsequently divided into two shape classes containing 167 (Fig. 5(b)) and 125 (Fig. 5(c)) spikes.

Table 1. Cluster size occurrence

Shapes detected	*Amplitude based*	*Wavelet based*
1	9	15
2	73	62
3	11	17
4	2	2
5	1	0



(a) All spikes (b) Class A spikes (c) Class B spikes

Fig. 5. An example of the spike discrimination: a) 312 spikes, b) 167 spikes, c)125 spikes

4 Power Spectrum Analysis

The area of the brain in which electrode should be inserted (STN) is characterized by high neuronal activity. This activity should be reflected in power of the signal. The raw recorded signal is highly contaminated with noise from neurons that are near the electrode. In [7], authors basing upon spikes occurrence in original signal create new one to conduct the synchronization analysis. In this paper similar procedure is used to create the temporary signal from the spikes and then analyze it's power using FFT.

4.1 Creating the Temporary Signal

The temporary signal has the same length as the original one. Its sample rate is always 1KHz. This sample rate according to Nyquist-Shannon sampling law ensures that frequencies up to 500Hz will be well described. Signal is created with constant amplitude 0, then at points that correspond to spike occurrences, a part of cosine function is inserted. Cosine function values are inserted in such a way that if the spike occurred at time t_0 then a part of cosine defined on $\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$ is mapped onto $< t_0 - 5ms, t_0 + 5ms >$. Mapping is done in additive way - if two or more spike induced cosines overlap they amplitudes summarize.

4.2 Extracting the Power Spectrum

When all spikes have their representation in the temporary signal, it is transformed using FFT to obtain the power spectrum. It is possible to obtain power spectrum for frequencies up to 500Hz. Power of the frequencies above 100Hz is very small and as frequency increases it quickly approaches zero. Because of that, only for frequencies less or equal 100Hz power spectrum is being observed, power of higher frequencies is discarded. Power for frequencies below 1Hz is also not taken into account, it comes from all spikes being separated from each other by 1s or more and not STN specific. Summarizing, power is calculated for frequency range from 1Hz to 100Hz with resolution 1Hz.

4.3 Power Analysis

In DBS surgery several electrodes traverse selected hemisphere on parallel trajectories towards the STN. Electrodes record potentials at the same time and for the same time period. It is safe to compare the power of the signal recorded in these electrodes. Assume that for a given electrode e the power of a signal recorded at depth d, calculated for frequency f is represented by $pwr\,(e,\ d,\ f)$. This cumulative power can be defined as shown by equation (9)

$$pwr_{cumul}\,(e,\ d) = \sum_{d_i \leq d} \sum_{f=1}^{100} pwr(e, d_i, f) \tag{9}$$

4.4 Usefulness of Cumulative Power

Cumulative power have some interesting properties that can be useful to neurosurgeons and neurologists.

Test data
The dataset contain recordings taken from 11 DBS surgeries. During surgeries there were 20 sets of the microelectrode recordings. Each set contained from 2 up to 4 microelectrodes; total 60 probing microelectrodes were used. In all sets, neurologists have selected one of the electrodes as trajectory for implantation of final, stimulating electrode.

Selecting electrode that will reach the STN with good accuracy
When the microelectrodes reach the estimated from MRI depth at which STN should be found, it is time to pick one of them as a trajectory for the final stimulating electrode. If at this final depth, a cumulative power is calculated for each of the scanning electrodes, then obtained values can be used to determine position of the DBS electrodes. Microelectrodes with higher value of cumulative power are far more likely to be the ones that actually have reached the STN. The cumulative power has been calculated for all (60) microelectrodes from our dataset.

If highest cumulative power was used as the criterium for selecting electrode from a given probing set, then: 13 out of 20 good electrodes would correctly be selected ($TruePositive$), 33 out of 40 wrong electrodes would correctly be labeled

($TrueNegative$). In 6 cases algorithm would falsely select wrong electrode as a good one ($FalsePositive$) and finally in 8 cases it would falsely label good electrode as a wrong one ($FalseNegative$). In case of 46 out of 60 (76.7%) electrodes the method would correctly label electrodes as good or wrong. The remaining 14 (23.3%) electrodes would be labeled wrongly. Specificity is $\frac{33}{33+6} = 0.85$, sensitivity is $\frac{13}{13+8} = 0.62$. See Table 2(a). In four sets, the electrode chosen by neurologist has 2^{nd} highest cumulative value. If highest or 2^{nd} highest cumulative power were used as criterium for selecting electrodes from a given probing set, then 17 out of 20 good electrodes would be correctly selected. Specificity is 0.85, sensitivity is 0.71. See Table 2(b).

Table 2. Classification results

(a)

	positive	negative	%
true	13	33	76.7%
false	6	8	23.3%

(b)

	positive	negative	%
true	17	33	83.3%
false	6	4	16.7%

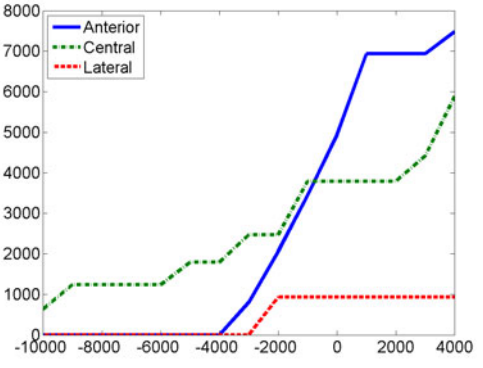


Fig. 6. Changes in cumulative power of signals simultaneously recorded from three microelectrodes over the depth

Predicting if electrodes are likely to reach the STN or not

It is desirable to know, as quickly as possible, if a given micoelectrode is going to reach the STN or not. If we know that a given microelectrode has minimal chance to reach the target, a neurosurgeon would not have to advance it deeper in the brain decreasing this way chances for additional side effects. Fig. 6 shows that already at the depth -1000 (1mm above estimated target position) one may suspect that *Anterior* microelectrode might be the best one (highest steepness) and that *Lateral* microelectrode will most probably miss the target.

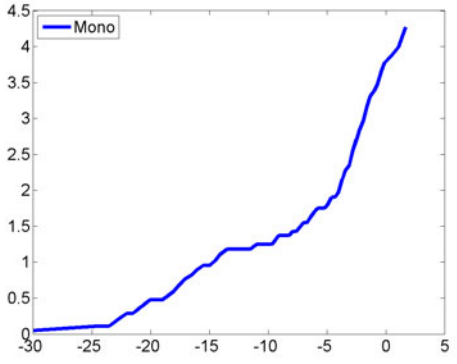


Fig. 7. Changes in cumulative power signal recorded from single microelectrode over the depth

Pinpointing depth of the microelectrode which reached the STN
In some cases only one probing electrode is inserted into patient's brain. It is then impossible to compare it with other data. Still the cumulative power gives us some information regarding whether and when electrode reached the STN. Fig. 7 shows that from depth of -5, the power of the signal steeply increases. With hight probability this is the depth about which STN has been reached.

5 Conclusions and Acknowledgement

We propose that the spike shape extraction, classification and their cumulative power spectra are new tools that might help to determine the exact STN coordinates. Our decision algorithm will increase surgery safety and improve precision of the STN stimulation that will make the DBS therapy more efficient. Computations were performed on Intel 3.33Ghz Windows 7 64bit machine. Software used: C .Net, Oracle 11g and Matlab R2009. Full analysis of single patient on average took below 10 minutes to complete.

The dataset containing recordings from 11 DBS surgeries has been provided by Dr. Dariusz Koziorowski from Bródnowski Hospital in Warsaw.

References

1. Henze, D.A., Borhegyi, Z., Csicsvari, J., Mamiya, A., Harris, K.D., Buzsák, G.: Intracellular Features Predicted by Extracellular Recordings in the Hippocampus In Vivo. Journal of Neurophysiology 84, 390–400 (2000)
2. Pettersen, K.H., Einevoll, G.T.: Amplitude Variability and Extracellular Low-Pass Filtering of Neuronal Spikes. Biophysical Journal 94, 784–802 (2008)

3. Bédard, C., Kröger, H., Destexhe, A.: Modeling Extracellular Field Potentials and the Frequency-Filtering Properties of Extracellular Space. Biophysical Journal 86, 1829–1842 (2004)
4. Wiltschko, A.B., Gage, G.J., Berke, J.D.: Wavelet Filtering before Spike Detection Preserves Waveform Shape and Enhances Single-Unit Discrimination. J. Neurosci. Methods 173, 34–40 (2008)
5. Archer, C., Hochstenbach, M.E., Hoede, C., Meinsma, G., Meijer, H.G.E., Ali Salah, A., Stolk, C.C., Swist, T., Zyprych, J.: Neural spike sorting with spatio-temporal features. In: Proceedings of the 63rd European Study Group Mathematics with Industry, January 28-February 1 (2008)
6. Quian Quiroga, R., Nadasdy, Z., Ben-Shaul, Y.: Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering. MIT Press, Cambridge (2004)
7. Levy, R., Hutchison, W.D., Lozano, A.M., Dostrovsky, J.O.: High-frequency Synchronization of Neuronal Activity in the Subthalamic Nucleus of Parkinsonian Patients with Limb Tremor. The Journal of Neuroscience 20, 7766–7775 (2000)

Biometric System for Person Recognition Using Gait

Marcin Derlatka

Bialystok University of Technology,
Wiejska Street 45C, 15-351 Bialystok, Poland
mder@pb.edu.pl

Abstract. This paper presents a practical approach to person recognition by gait. The biometric system is based on ground reaction force (GRF) and positions of the center of pressure (COP) generated during the subject gait and the single-output multilayer neural network. The article discusses both the identification and the verification problems as well as influence of the security level on the quality of the proposed biometric system. The achieved results (more than 92% of correct recognition) show that human gait is a biometric measure which enables efficient authorization in a simple way. It could be used as a security system of a limited number of registered users.

Keywords: biometrics, human gait, force plate, neural networks.

1 Introduction

The biometrics (technical biometrics) can be regarded as a measurable psychological or behavioral characteristics of the individual, which is applicable in personal identification and verification. Using the most commonly biometrics authentication methods such as fingertips[7], face[2], iris[4], hand geometry[11], retina[13], gait[5] or mixed methods[8] have the significant advantage over traditional authentication techniques based on, for instance, the knowledge of passwords or the possession of a special card. The biometric patterns are unique for each individual and cannot be lost, stolen or forgotten. This is the reason why biometrics systems are recently increasingly popular.

Among the above given biometrics methods the special attention should be paid to human gait. Gait is a very complex human activity. It is a symmetrical and repetitive phenomenon in its normal form. Human gait could be used as a biometric measure because it is:

- a common phenomenon;
- measurable;
- unique for every person - the gait pattern is formed before a child is seven;
- quite unchangeable.

Moreover, in contrast to other biometrics authentication methods, human gait has the following advantages:

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 565–574, 2011.

- the person does not need to interact with any measurement system in an unnatural way; it is sufficient that the subject passes through a pathway equipped, in this case, with a force platform;
- trails can be done only by a living person;
- the verified subject does not need to be aware of being submitted to the authentication procedure.

Human gait could be described by an enormous number of parameters which include:

- kinematics;
- kinetics;
- anthropometrics;
- electromyographics;
- others.

Because it is impossible to investigate all gait parameters this paper will focus on data obtained by means of a force plate - ground reaction force (GRF) and the position of center of pressure (COP). The selection of parameters have been made on the criteria that the gait parameters should be easily measured outside the laboratory and should be characteristic for a person.

Gait is not actually new biometrics. Nowadays the most popular approach used in authentication people by the way they walk is based on a video analysis[10]. In this case, a set of parameters is calculated from a sequence of images both model-based and model-free. In the model-based approach the set of numbers reflects movements of a human body[3]. In the model-free approach the numbers are often derived from the sequence of silhouettes[1]. The problems with using video data in outdoor biometrics systems are: changes of the natural lighting and the clothing of the investigated person. In [9] footprints of the individuals have been used for human recognition. In this unique approach the authors achieved up to 85% of correct recognition. A different approach has been proposed in[12]. The authors used the vertical component of the ground reaction force (GRF) and the nearest neighbour classification for subject recognition. They used ten parameters such as: the mean and standard deviation of the profile, the length of the profile (number of samples), the area under the profile and finally the coordinates of two maximum points and the minimum point to describe each of the GRF profiles.

This paper describes the biometric system which made the authentication task based on the signals obtained by a force platform by means of set single-output multilayer neural networks. The main aim of this paper is to show that the basic transformations of GRF signal provide a great opportunity of using human gait as quite powerful biometrics.

2 The Biometric System Using Human Gait

There are two main tasks to be performed by the biometric system. They are: verification and identification. The system in the identification task should return

the name or the number of the investigated subject. In the verification task the user introduces himself and enters his biometrics patterns. The system should check if the presented pattern does belong to the user. The biometric system could be regarded as a classifier which maps the input vector into the class identifier. In this approach the number of classes in the verification task is equal to the number of the verified subjects. In the case of the identification task the number of classes should be higher by one than in the verification task, because the additional class is needed for the person who is not present in the database.

2.1 Ground Reaction Force and Center of Pressure

In the biomechanical approach, the ground reaction force (GRF) is the force which is acting on the body as a response to its weight and inertia during the contact of the human plantar with the surface. The time when the human plantar contacts the surface is called the support phase of gait[14]. The all three components of GRF were used in the presented work. They were: anterior/posterior F_x, vertical F_y and medial/lateral F_z components of GRF. The common profiles of the GRF components are presented in Fig. 1 (a-c).

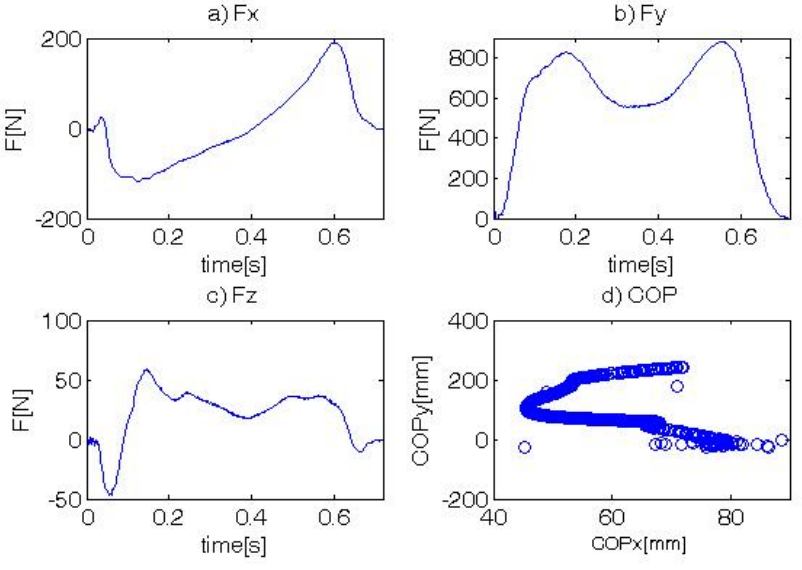


Fig. 1. Analyzed signals: components of GRF in: a) anterior/posterior, b) vertical, c) medial/lateral direction, d) trajectory of COP during the support phase

The anterior/posterior component has two main phases. The value of F_x is negative in the first phase. It is a result of the deceleration of the investigated lower limb, in this case the force direction is opposite in direction of walking.

The minimum of the deceleration phase is most often reached a moment before the maximum of the limb-loading phase in the vertical component of GRF. The value of F_x is positive in the second phase, respectively. The maximum of the acceleration phase is reached when the toe-off phase starts.

There are three extremes in Fig. 1b. They correspond to:

- the maximum of the limb-loading phase;
- the minimum of the limb-unloading phase;
- the maximum of the propulsion phase (a moment before the toe off).

It is not difficult to point to the same extremes for the medial/lateral component of GRF as for the vertical GRF.

It is important to note that values and profile of GRF depend on the velocity of walking and body weight of the investigated subject.

A force plate could measure the center of pressure (COP), too. The COP is the point of location of the vertical component of GRF. The point is given as coordinates where the origin of coordinates is determined before the experiment.The work presented took into consideration the trajectory of COP during the support phase of the subject. In [6] COP has been used as a good index to calculate the balance of individuals.

2.2 Feature Extraction

All three components of GRF and the coordinates of COP position give five vectors of the measure values depended on time as a result of the investigation. The measured data can be presented as five vectors of the same length:

$$p^i = \begin{bmatrix} \mathrm{p}_1^i \\ \mathrm{p}_2^i \\ \vdots \\ \mathrm{p}_N^i \end{bmatrix} \tag{1}$$

where i=1,2,3,4,5, denotes token's parameters: coordinates of COP (COP_x and COP_y), and all components of GRF: F_x, F_y and F_z; N is the number of the measuring points.

Based on the vector above, a matrix which is representation of one token can be created:

$$\mathrm{P} = \begin{bmatrix} \mathrm{p}_1^1 & \mathrm{p}_1^2 & \cdots & \mathrm{p}_1^5 \\ \mathrm{p}_2^1 & \mathrm{p}_2^2 & \cdots & \mathrm{p}_2^5 \\ \vdots & \vdots & & \vdots \\ \mathrm{p}_N^1 & \mathrm{p}_N^2 & \cdots & \mathrm{p}_N^5 \end{bmatrix} \tag{2}$$

Here, in contrast to the approach used in biomechanics, the GRF is not normalized, because the value of GRF corresponds to body weight of the investigated subject, so it could be useful in distinguishing individuals. In the presented work the following parameters are extracted for representation of the support phase of a single step:

- duration of the support phase;
- mean of the each GRF profiles;
- standard deviation of the each GRF profiles;
- eigenvalues and eigenvectors of the covariance matrix created based on token P (only parameters indicated by i=3,4,5);
- the coefficients of a polynomial of 5th degree that fits the F_j=f(time) best in a least-squares sense: $a_{j,5}$,$a_{j,4}$, $a_{j,4}$, $a_{j,3}$, $a_{j,2}$, $a_{j,1}$, $a_{j,0}$, where $j \in \{x, y, z\}$;
- the coefficients of a polynomial of 5th degree that fits the COP_x=f(COP_y) best in a least-squares sense b_5, b_4, b_3, b_2, b_1 except b_0, because b_0 don't indicate the shape of COP_x=f(COP_y), but only position of heel strike during trials.

As a conclusion, each token is converted into an input vector used by the neural network in the biometric system.

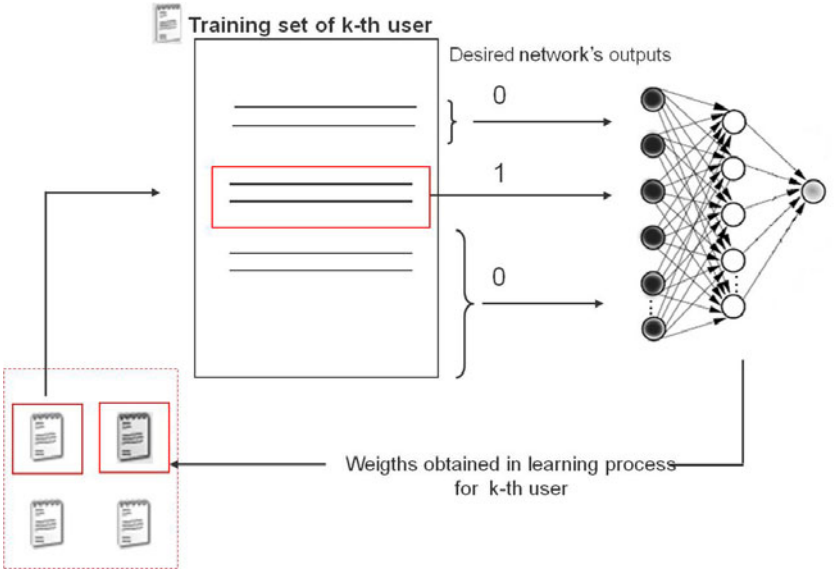


Fig. 2. The process of learning the single user gait

2.3 Applying of the Neural Networks

The classical multilayer neural network with one hidden layer and one output neuron is implemented. Each neuron has a sigmoid activation function. The neural network is learned with Rprop algorithm with the same learning parameters as many times as the users are presented in the database. The network is learned to recognize only one user at time (Fig. 2). It is achieved by changing the network output desirable value in the training set. The neural network is learned to get the output equal 1 for the considered user and to get output equal 0 for other users. The neural network is initialized with the same weight's values before the

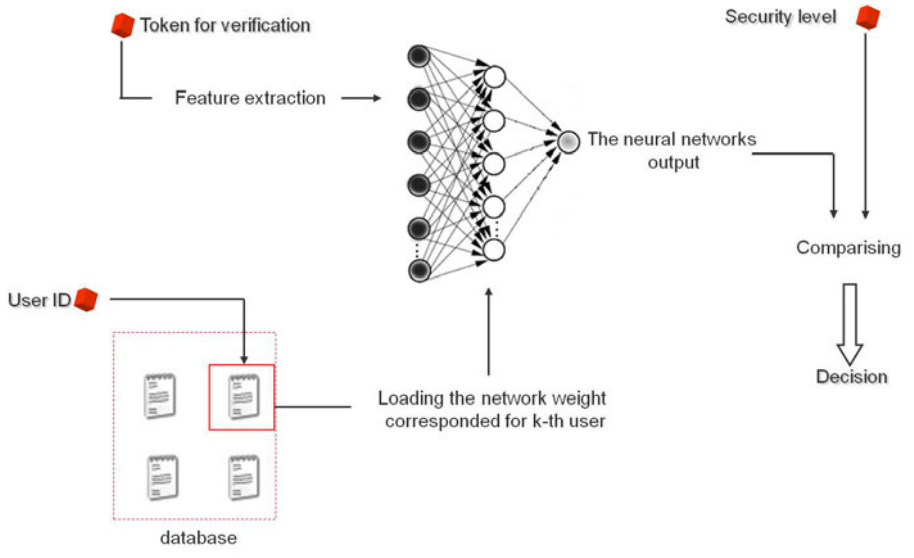


Fig. 3. The process of verification the single user

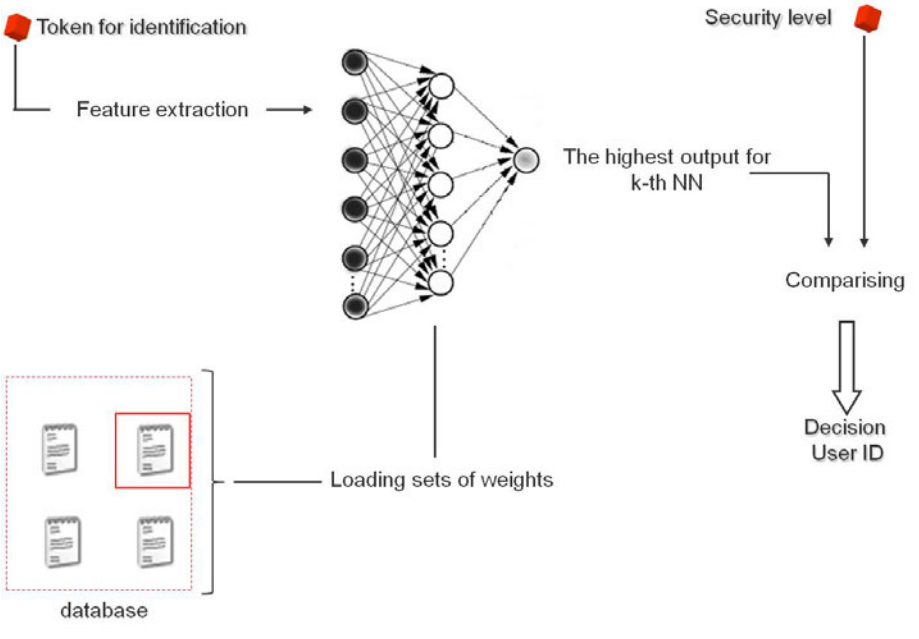


Fig. 4. Scheme of identification process

learning process for each user. The weights of the neural networks are written into the proper file associated with the considered user after training.

In the verification task the neural network weights are read from the file associated with the claim user. The user's gait is transformed according to the way described in subsection 2.2 and the input vector is presented into the network's input. The response value is compared with the assumed security level ($0.1 \div 0.9$). The positive verification is accepted only if the network response is greater than the security level (Fig. 3).

In the identification task, as well as in the verification the user's gait is transformed into the network's input vector. Subsequently, the network response for every weights vector stored in the database is checked up. The system chooses this user's ID for which the neural network response is the greatest. If the chosen response of the neural network is greater than the assumed threshold the decision about the recognition of a user is positive. Otherwise the system treats the subject as unrecognized (Fig. 4).

3 Results

3.1 Experimental Population

The measurements were made in the Bialystok University of Technology on a group of 25 volunteers (17 men and 8 women) by means of the Kistler force plate. The subjects who took part in the investigations were at age 21.4 ± 0.59, body weight 77.79 ± 19.4 and body height 176.68 ± 10.65. Eighteen subjects had not experienced injuries or abnormalities affecting their gait. Seven volunteers reported a sprain of ankle in the past (from 1 year to 10 years before investigation). Among them was one subject who walked in an unnatural way with the hands at the front of his body. Moreover, one more subject had his right lower limb 3.5 cm longer than the left one.

The one Kistler force plate was hidden on the pathway and recorded data with frequency 1kHz. The volunteers made several trails ($10 \div 14$) with comfortable self-selected velocity in their own shoes hitting at the plate with the left or the right leg, so almost 300 steps have been recorded. 158 tokens from 20 users were used to build the learning set. 138 tokens from all 25 users were treated as the testing set. Five subjects (60 trials) who were represented only in the testing set were used for checking the biometric system's ability to stop intruders both in the identification and the verification tasks. Among the intruders only one reported a sprain of ankle injury which took place 5 years before the investigation. The rest of them did not rise any health problem which can influence their gait.

3.2 Recognition Accuracy

The results of the identification and verification of the subjects depending on the security level are presented in Table 1 and Table 2. The presented values were calculated on the base of data from the testing set.

Table 1. The results of the identification of the users and intruders tokens

Security level	Identification for users tokens		Identification for intruders tokens
	false rejected	false accepted	false accepted
0.1	0%	3.85%	53.33%
0.2	1.28%	2.56%	30.00%
0.3	2.56%	1.28%	18.33%
0.4	3.85%	1.28%	12.82%
0.5	5.13%	1.28%	6.67%
0.6	5.13%	0%	5.00%
0.7	5.13%	0%	1.67%
0.8	5.13%	0%	0%
0.9	5.13%	0%	0%

In the identification task the real problem is the very high percentage of intruders false accepted tokens for small values of the security level. In fact, it determines that one should takes into consideration the threshold equal or greater than 0.6. It is easy to notice that for the security level 0.6 and greater there is no change in percent of false rejected tokens and of false accepted tokens for the subjects who has the representation of their gait in the training set (users). It leads to conclusion that for the presented biometric system using gait the best results of subject recognition can be achieved for the security level equal 0.8.

The more detailed analysis of the results indicates that there are no special problems with identification the users who had sprained ankle injuries in the past. The biggest problem in the identification generates tokens of the subject with unequal length of legs. This needs a low value of security level to be recognized by the biometric system in the case of his two (from all four) tokens. One of that problematic tokens corresponds to the left lower limb and the second corresponds to the right lower limb. In fact, such a big difference in length between the left and the right leg leads to an unsymmetrical gait and differences in the patterns describing only one lower limb. Consequently, this rises a need for proceeding the measures for both legs simultaneously by means of two force plates.

The results of the verification demonstrate that the intruders have really small chances to be accepted by the biometric system as somebody who the system knows. On the other hand, it is rather high percentage (7.69% for the security level equal or bigger than 0.5) of false rejected tokens for the users. There is not a very big problem to repeat the gait trail, but it should be underlined that frequent repetitions generate frustration of the users.

To sum up, the general recognition accuracy is really high (more than 92% for the security level equal or greater than 0.5). However, we should remember that the recognition accuracy depends on the task and the signals used while comparing with the results of the other authors who reported the application human gait in the biometric system. The results reported in the work [12] where

Table 2. The results of the verification in respect of the users and intruders tokens

Security level	Verification for users tokens	Verification for intruders tokens
	false rejected	false accepted
0.1	2.56%	4.08%
0.2	3.85%	1.75%
0.3	5.13%	0.92%
0.4	6.41%	0.5%
0.5	7.69%	0.33%
0.6	7.69%	0.25%
0.7	7.69%	0.08%
0.8	7.69%	0%
0.9	7.69%	0%

GRF profiles have been used, are in the range from 16% up to 93% correct recognition. However it needs underlining that in [12] the subjects had been instructed to place their foot in the center of the used device. As a result the authors recorded and proceeded the unnatural gait.

4 Conclusions

The presented biometric system based on the signals derived from a force plate shows that the human gait is a biometric measure which enables efficient authorization in a simple way. It works in the way invisible to the subjects. It can be successfully applied in both indoor or outdoor conditions without any changes and, what is the most important, it works with quite high efficiency. The author is aware of some limitations of the presented results, of course. First of all, the results show the need for investigation based on two force plates. Only this approach will eliminate the problem with recognizing users with unsymmetrical gait. Second, the presented system is hard flexible. It is necessary to rebuild each of the training sets and to retrain each of the neural networks in the case of addition at least one more authorized user to the database. Third, the results obtained are not based on a large database, so the question of the system's scalability is still open. However, it is worth emphasizing that in this work the voluntaeers have been recruited from students. Thus, the database contains more homogenic (that is harder for proper classification) patterns then in real life. These limitations will be dealt with in future works.

Acknowledgments

This paper is supported by grant S/WM/1/09 from the Bialystok University of Technology.

References

1. Balista, J.A., Soriano, M.N., Saloma, C.A.: Compact Time-independent Pattern Representation of Entire Human Gait Cycle for Tracking of Gait Irregularities. Pattern Recognition Letters 31, 20–27 (2010)
2. Chan, L.H., Salleh, S., Ting, C.M.: Face Biometrics Based on Principal Component Analysis and Linear Discriminant Analysis. Journal of Computer Science 6(7), 691–698 (2010)
3. Cunado, D., Nixon, M.S., Carter, J.N.: Automatic Extraction and Description of Human Gait Models for Recognition Purposes. Computer Vision and Image Understanding 90(1), 1–41 (2003)
4. Daugman, J.: How iris recognition works. IEEE Transactions on Circuits and System for Video Technology 14(1), 21–30 (2004)
5. Goffredo, M., Bouchrika, I., Carter, J.N., Nixon, M.S.: Self-Calibrating View-Invariant Gait Biometrics. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 40(4), 997–1008 (2010)
6. Karlsson, A., Frykberg, G.: Correlations Between Force Plate Measures for Assessment of Balance. Clin. Biomech. 15(5), 365–369 (2000)
7. Lin, C.H., Chen, J.L., Tseng, C.Y.: Optical Sensor Measurement and Biometric-Based Fractal Pattern Classifier for Fingerprint Recognition. Expert Systems with Applications 38(5), 5081–5089 (2011)
8. Liu, Z., Sarkar, S.: Outdoor Recognition at a Distance by Fusing Gait and Face. Image and Vision Computing 25, 817–832 (2007)
9. Nakajima, K., Mizukami, Y., Tanaka, K., Tamura, T.: Footprint-Based Personal Recognition. IEEE Transactions on Biomedical Engineering 47(11), 1534–1537 (2000)
10. Nash, J.N., Carter, J.N., Nixon, M.S.: Extraction of Moving Articular-Objects by Evidence Gathering. In: Proc. of the 9th British Machine Vision Conference, pp. 609–618 (1998)
11. Niennattrakul, V., Wanichsan, D., Ratanamahatana, C.A.: Hand geometry verification using time series representation. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part II. LNCS (LNAI), vol. 4693, pp. 824–831. Springer, Heidelberg (2007)
12. Orr, R.J., Abowd, G.D.: The Smart Floor: a Mechanism for Natural User Identification and Tracking. In: Proc. of Conference on Human Factors in Computing Systems (2000)
13. Usher, D., Tosa, Y., Friedman, M.: Ocular Biometrics: Simultaneous Capture and Analysis of the Retina and Iris. Advances in Biometrics 1, 133–155 (2008)
14. Winter, D.A.: Biomechanics and Motor Control of Human Movement, 4th edn. John Wiley & Sons Inc., Chichester (2009)

minedICE: A Knowledge Discovery Platform for Neurophysiological Artificial Intelligence

Rory A. Lewis[1,2] and Allen Waziri[3]

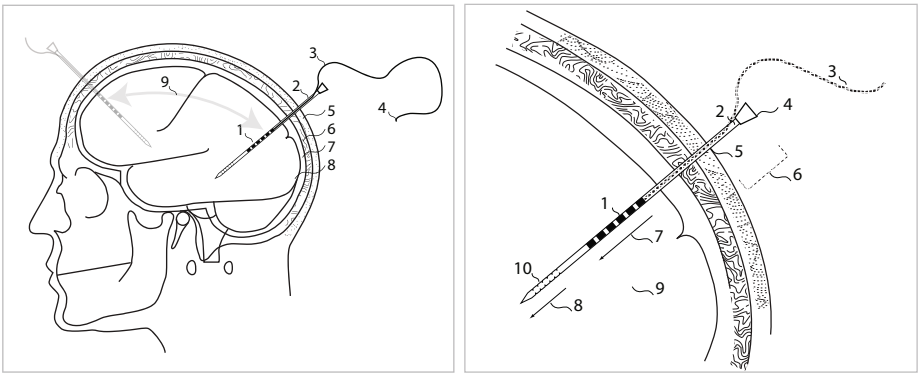
[1] Department of Computer Science, University of Colorado at Colorado Springs, Colorado Springs, CO, 80933
[2] Departments of Pediatrics & Neurology, University of Colorado Denver, Anschutz Medical Campus, Denver, CO 80262
[3] Department of Neurosurgery, University of Colorado Denver, Anschutz Medical Campus, Denver, CO 80262

Abstract. In this paper we present the minedICE^{TM} computer architecture and network comprised of neurological instruments and artificial intelligence (AI) agents. It's called *minedICE* because data that is "*mined*" via **I**ntra**C**ortical **E**lectroencephalography (*ICE*) located deep inside the human brain procures (*mined*) knowledge to a Decision Support System (DSS) that is read by a neurosurgeon located either at the bedside of the patient or at a geospatially remote location. The DSS system 1) alerts the neurosurgeon when a severe neurological event is occurring in the patient and 2) identifies the severe neurological event. The neurosurgeon may choose to provide feedback to the AI agent which controls the confidence level of the association rules and thereby teaches the learning component of minedICE.

1 Introduction

The detection and interpretation of abnormal brain electrical activity in patients with acute neurological injury remains an area of significant opportunity for technological advancement. Neurosurgeons know that when a patient arrives in the emergency room (ER) with a severe head injury, they rely on their intuition and relatively limited external data to choose the necessary treatment modality. Aside from the initial injury, the brain tissue in these patients is extremely susceptible to secondary injury from ongoing abnormal (and preventable) physiological processes. Although a number of invasive neuromonitoring systems exist, current modalities either provide indirect measurement of brain health (and are therefore difficult to interpret) or have limited sensitivity and specificity for accurately identifying critical and deleterious changes in brain health. To overcome this limitation, Waziri developed Intracortical Electroencephalography (ICE) [10] a technique by which specialized multicontact electrodes can be placed into the cerebral cortex through a burrhole generated at the bedside, as illustrated in Figure 1. Through the use of ICE, high amplitude and high fidelity EEG data can be recorded in an otherwise electrically noisy environment.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 575–580, 2011.



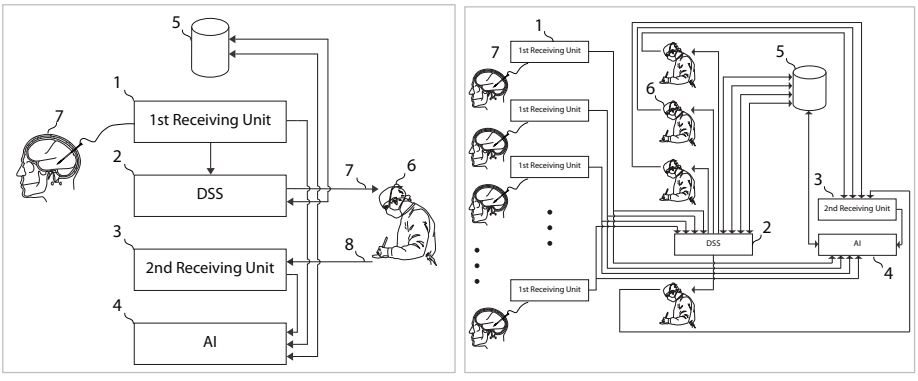
(a) (b)

Fig. 1. *Intracortical Electroencephalograph (ICE) Pin, Side Elevations:* ((a) ICE ***2*** inserted through periostem ***5***, skull ***6***, arachnoid and pia mater ***7*** into brain ***8*** in varying positions ***9***. Receiving electrodes ***1*** encapsulated by brain ***8***. Electrodes ***1*** transmit signals along wire ***3*** to end ***4*** where it is connected to computer. (b) Cannula and internal lumen ***2*** with drainage hole ***4*** and sharpened end ***8***. Electrodes ***1*** at electrode region ***17*** allow insertion through burr hole ***5*** traversing brain ***9***. External region ***6***of cannula remains outside of skull. Connection conductors ***7*** combine into a single wire ***2***. Drainage holes ***10*** in drainage region ***8*** provide openings for fluid to flow ***4***. Support member inserted through ***4*** into internal lumen for accurate placement).

1.1 minedICE Architecture

The authors have tested systems on humans and pigs using Weka, Matlab, RSES and TunedIT to run KDD techniques in the initial interpretation of EEG data. Herein we present a system that, from a high level, dynamically reads and converts EEGs into the time and frequency domains where it compares them to a database and then instructs a DSS to tell the neurosurgeon how confident it is that a particular neurological event is occurring (see Figure 3). If the surgeon provides feedback, it updates the association rules and confidence algorithms making it more intelligent for the next patient. The basis of the architecture has been the author's work using deterministic finite automata [8], [4], [5], [6]. Now the authors move on to detecting life threatening neurological events [7].

Figure 2 §(a) represents a single patient-to-neurosurgeon view and §(b) one of many ways a hospital could link multiple patients to multiple neurosurgeons. Signals received at the 1st receiving unit *1* are channeled in real time to two arrays, one in time domain and the other in the frequency domain. A discrete finite automata module segments critical areas for the discretization units autonomous to each of the two time and frequency streams. A first stream of discretized data is compared to a database that contains association rules where an identity ζ, number π located at frequency μ has a confidence of σ. Accordingly, each $(\zeta_{\mu}^{\pi})^{\sigma}$ is passed 1) to update the AI module and 2) to the DSS unit. DSS interpolation: The plurality of $(\zeta_{\mu}^{\pi})^{\sigma}$ are associated with an ontology in the DSS *2* module



(a) (b)

Fig. 2. *Architecture:* ((a) Local architecture: 1st Receiving Unit ***1*** DSS ***2***, 2nd Receiving Unit ***3***, AI module ***4*** central database ***5*** neurosurgeon ***6*** and patient ***7***. DSS connection to Neurosurgeon ***7*** and AI receiver from neurosurgeon ***8***. (b) Hospital WAN: Plurality of 1st Receiving Units ***1*** single DSS server***2***, single 2nd Receiving Unit server ***3***, AI module ***4*** central database ***5*** neurosurgeons ***6*** and patient ***7***. DSS connection ***7*** AI receiver ***8***).

which alerts and tells the neurosurgeon *6* the state and the probability of that data mined neurological state. The 2nd Receiving Unit *3* receives feedback from the neurosurgeon which converts various forms of input back to the $(\zeta_{\mu}^{\pi})^{\sigma}$ format. Machine Feedback: Utilizing an unsupervised algorithm by the author illustrated in [3] the difference in the confidence level of each σ in $(\zeta_{\mu}^{\pi})^{\sigma}$ [11], is $\phi(x)$ for each number $\pi \in score$. It is only at this point the database *5* is updated and the associated rules relying on the new $(\zeta_{\mu}^{\pi})^{\sigma}$ in the 1st receiving unit *1* are updated.

2 Experiments

In order to separate and classify each significant portion of the streaming ICE signal π in terms of the Fast Fourier Transform (FFT) coefficients of the artifact polluting the signal we extract the features of the ICE 4 times for each threshold π before using classical KDD tools as illustrated in Figure 4 to identify each relevant ζ^{π} where π is compared when $(\pi \in 1, 2.5, 5....10)$ for $n = 256$ and $(\pi \in 1, 2.5, 5....10)$ for $n = 2048$. Next we divide each signal π of the set of selected signals Π into equal-sized non-overlapping hops with size $2n$ samples $\Theta^{\pi} = \theta_1(\pi), \theta_1(\pi), \theta_2(\pi), ..., \theta_r(\pi)$ where r is the size of $\frac{(\pi)}{2n}$ Once this is repeated for both $n = 256$ and $n = 2048$ for each signal π we pick up hops $\theta_i(\pi)(1 <\leq 1 \leq)r$ such that s hops for π form the set Γ^{π} $(\Gamma^{\pi} \subseteq (\Theta^{\pi}(\Gamma^{\pi} = (\gamma_1(\pi), \gamma_2(\pi), ..., \gamma_r(\pi))))$. Now that we have picked up our significant hops we perform FFTs on them such that the amplitude of the complex portion is calculated and stored as a pointer. A simple aggregation loop is them performed at

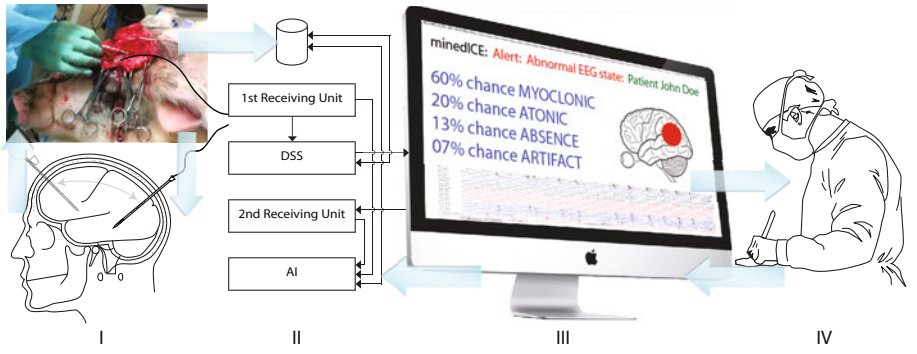


Fig. 3. *Interim Architecture:* I. Learn classifiers between pig / human neurological states. II. 1st Receiving unit sends discretized signal to AI and DSS. III. minedICE procures DSS to neurosurgeon. IV. Neurosurgeon provides feedback.

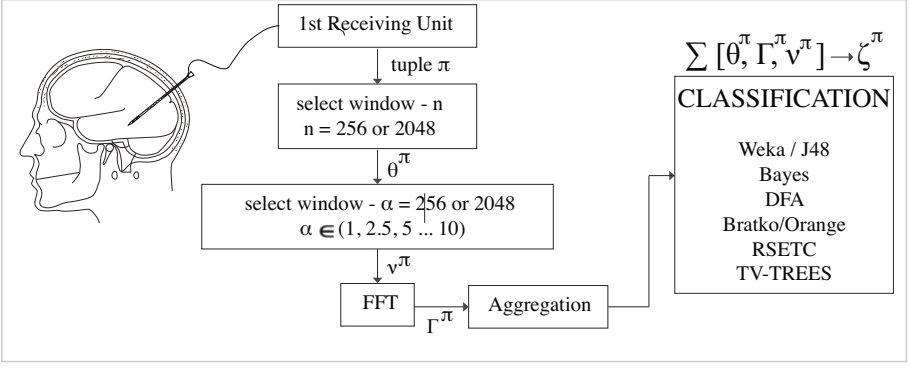


Fig. 4. Receiving Unit to Classification

each pointer as they turn up in the system. The results are now ready for our KDD experiments.

To test the feasibility of the various KDD Tools we used C++ DFA1, SVM Weka/J48, Bayes, DFA, Bratko/Orange and RSETC to build models for each of 9 Reports, made for the purpose of this test wherein the authors provided 9 test junctures in signals wherein the overall curve of the DFA tree resembled an arc. These reports were named according to the Power spectrum at each point. As shown in Figure 5 one sees the Report 11010 etc which acted as the training data and we chose the J48 decision tree as our classification algorithm. To test how the Weka instantiated a tree we found that it did break off at 9 leaves with 12 branches. We were not able to get Orange to output a similar tree. Looking at Figure 5 we see that the Weka /J48 system correlates closest to the training set's general arc. It is interesting and certainly cause for investigation why SVM hsot up, rather than down at Report11011. It is also interesting that DFA, C++

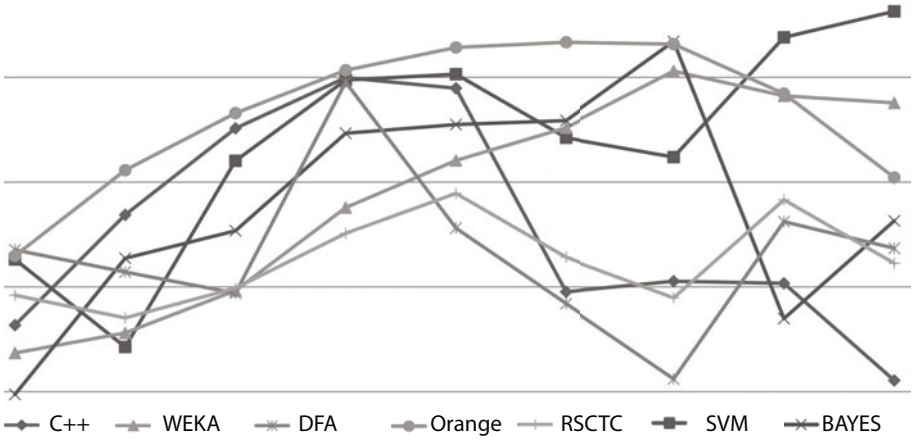


Fig. 5. Receiving Unit to Classification

DFA1, RSES and Bayes are clumped together after Report11011 in the 1,000 to 2,000 range. It is also interesting that at Report 11110, apart from Bayes, Bratco/Orange, and C++ DFA 1, all the KDD tool nailed it on 2,500 perfectly.

3 Conclusion and Future Work

The system is able to datamine a stream of signals and pickup the correct patterns from within FFT's. This is the good news. The bad news is that these tools' results should have been much closer. Our future work will be to analyze why the disparities of the experiments existed. Also we were not able to match the data in a form to work with Action Rules as in the past [1], [9]. This may turn out to be crucial because as the crucial element in minedICE is that the system must learn [2]. As the system learns and makes new rules some of the form of Action Rules or TV Trees may be necessary to manipulate the tree for the DFA. We may find that Weka/J48 is not the best when we correct out possible errors. We may also find that out methodology of converting the signals into FFTs for the classification module is inherently flawed. Essentially we need to run these tests a lot more and make the system more robust. Again though, we know that as each set of tests are concluded the machine is getting to the point that it will read actual human data and possible save lives.

References

1. Lewis, R., Raś, Z.: Rules for Processing and Manipulating Scalar Music Theory. In: Proceedings of the International Conference on Multimedia and Ubiquitous Engineering, MUE 2007, April 26-28, pp. 819–824. IEEE Computer Society, Los Alamitos (2007)

2. Lewis, R., Cohen, A., Jiang, W., Raś, Z.W.: Hierarchical tree for dissemination of polyphonic noise. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 448–456. Springer, Heidelberg (2008)
3. Lewis, R., Kalita, J., Sarmah, S., Bhattacharyya, D.: Music Industry Scalar Analysis Using Unsupervised Fourier Feature Selection. In: Proceedings of IIS 2009, Recent Advances in Intelligent Information Systems, Krakow, Poland, June 15-18, pp. 562–571 (2009)
4. Lewis, R., Parks, B., Shmueli, D., White, A.M.: Deterministic Finite Automata in the Detection of Epileptogenesis in a Noisy Domain. In: Proceedings of the Joint Venture of the 18th International Conference Intelligent Information Systems (IIS) and the 25th International Conference on Artificial Intelligence, Siedlce, Poland, June 8-10, pp. 207–218 (2010)
5. Lewis, R., Parks, B., White, A.M.: Determination of Epileptic Seizure Onset From EEG Data Using Spectral Analysis and Discrete Finite Automata. In: Proceedings of the 2010 IEEE International Conference on Granular Computing, Silicon Valley, August 14-16, page will appear (2010)
6. Lewis, R., Parks, B., White, A.M.: Discrete Finite Automata and KDD for Mining EEG Spikes and Seizures. In: Proceedings of XVIIth International Conference on Systems Science, Warsaw, Poland, September 14-16 (2010) (will appear)
7. Lewis, R., White, A.M.: Multimodal Spectral Analysis and Discrete Finite Automata for Detecting Seizures. In: Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence (WI 2010) and Intelligent Agent Technology, IAT 2010, Toronto, Canada, August 31-September 3 (2010) (will appear)
8. Lewis, R., White, A.M.: Seizure Detection Using Sequential and Coincident Power Spectra with Deterministic Finite Automata. In: Proceedings of the International Conference on Bioinformatics and Computational Biology (BIOCOMP), Las Vegas Nevada, July 12-15, vol. II, pp. 481–488 (2010)
9. Lewis, R.A., Wieczorkowska, A.: Categorization of Musical Instrument Sounds Based on Numerical Parameters. In:Conceptual Structures: Knowledge Architectures for Smart Applications, in Proceedings of RSEISP LNAI, 15th International Conference on Conceptual Structures, ICCS 2007, vol. 4604, pp. 784–792. Springer, Heidelberg (2007)
10. Waziri, A., Claassen, A.J., Stuart, R.M., Arif, H., Schmidt, J.M., Mayer, S.A., Badjatia, N., Jull, L.L., Connolly, E.S., Emerson, R.G., Hirsch, L.J.: Intracortical electroencephalography in acute brain injury. Annals of Neurology 66(3), 366–377 (2009)
11. Wieczorkowska, A., Synak, P., Lewis, R., Raś, Z.: Creating Reliable Database for Experiments on Extracting Emotions from Music. In: Proceedings of the IIS 2005 Symposium on Intelligent Information Processing and Web Mining, Advances in Soft Computing, pp. 395–404. Springer, Gdansk (2005)

On Different Types of Fuzzy Skylines

Allel Hadjali[1], Olivier Pivert[1], and Henri Prade[2]

[1] Irisa – Enssat, University of Rennes 1
Technopole Anticipa 22305 Lannion Cedex France
[2] IRIT, CNRS and University of Toulouse, 31062 Toulouse Cedex 9, France
hadjali@enssat.fr, pivert@enssat.fr, prade@irit.fr

Abstract. This paper deals with database preference queries based on the skyline paradigm, which aim at retrieving the tuples non Pareto-dominated by any other. We propose different ways to fuzzify such queries in order to make them more flexible, to increase their discrimination power, to make them more drastic or more tolerant. In particular, some of these extensions make it possible to reduce the risk of getting many incomparable tuples, even when the number of dimensions is high.

1 Introduction

Numerous approaches have been proposed to make database systems more flexible in supporting user preferences (see [1] for a survey). One of the most well-known approaches is that of skyline queries proposed in [2]. Given a set r of n-dimensional tuples or points, a skyline query returns the set of non-dominated points in r. A tuple t dominates a tuple t' if t is at least as good as t' in all dimensions and strictly better than t' in at least one dimension.

Several research efforts have been made to develop efficient algorithms and to introduce different variants for skyline queries [3,4,5,6,7,8]. In particular, the problem of skyline rigidity is addressed in [9] where a flexible dominance relationship is proposed. It allows the enlarging of the skyline with points that are not much dominated by any other point (even if strictly speaking they are dominated). This issue is also addressed in [10] through an extension of the *winnow* operator initially proposed in [11]. However, many other ways to make skyline queries "fuzzy" can be thought of, and the objective of the present paper is to present and discuss some of them, that we think meaningful.

The paper is structured as follows. Section 2 consists of a reminder about skyline queries. Section 3 describes five different ways in which a skyline may become "fuzzy" when it is refined, relaxed, simplified, extended to uncertain data, or generalized to incompletely stated context-dependent preferences. Section 4 concludes the paper and outlines some perspectives for future research.

2 Reminder about Skyline Queries

The notion of a skyline in a set of tuples is easy to state (since it amounts to exhibit non dominated points in the sense of Pareto ordering). Assume we have:

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 581–591, 2011.

- a given set of criteria $C = \{c_1, \ldots, c_n\} (n \geq 2)$ associated respectively with a set of attributes A_i, $i = 1, \ldots, n$;
- a complete ordering $\succcurlyeq_i$ given for each criterion i expressing preference between attribute values[1] (the case of non comparable values is left aside).

A tuple $u = (u_1, \cdots, u_n)$ in a database D *dominates* (in the sense of Pareto) another tuple $u' = (u'_1, \cdots, u'_n)$ in D, denoted by $u >_{dom} u'$, iff u is at least as good as u' in all dimensions and strictly better than u' in at least one dimension:

$$u >_{dom} u' \Leftrightarrow \forall i \in \{1, \ldots, n\},\ u_i \succcurlyeq_i u'_i \text{ and } \exists i \in \{1, \ldots, n\} \text{ such that } u_i \succ_i u'_i. \quad (1)$$

A tuple $u = (u_1, \cdots, u_n)$ in a database D belongs to the skyline S, denoted by $u \in S$, if there is no other tuple $u' = (u'_1, \cdots, u'_n)$ in D which dominates it:

$$u \in S \Leftrightarrow \forall u',\ \neg(u' >_{dom} u). \quad (2)$$

Then any tuple u' is either dominated by u, or is non comparable with u. In the following, we denote by $Dm(u)$ those tuples from D that are dominated by u:

$$Dm(u) = \{u' \in D \mid u >_{dom} u'\} \quad (3)$$

and by $Inc(u)$ those tuples which are non comparable with u:

$$Inc(u) = \{u' \in D \mid u' \neq u \wedge \neg(u >_{dom} u') \wedge \neg(u' >_{dom} u)\} \quad (4)$$

Table 1. An extension of relation *car*

	make	*category*	*price*	*color*	*mileage*
t_1	Opel	roadster	4500	blue	20,000
t_2	Ford	SUV	4000	red	20,000
t_3	VW	roadster	5000	red	10,000
t_4	Opel	roadster	5000	red	8000
t_5	Fiat	roadster	4500	red	16,000
t_6	Renault	coupe	5500	blue	24,000
t_7	Seat	sedan	4000	green	12,000

Example 1. Let us consider a relation *car* of schema (*make*, *category*, *price*, *color*, *mileage*) whose extension is given in Table 1, and the query:

select * **from** car **preferring**
(*make* = 'VW' **else** *make* = 'Seat' **else** *make* = 'Opel' **else** *make* = 'Ford') **and**
(*category* = 'sedan' **else** *category* = 'roadster' **else** *category* = 'coupe') **and**
(**least** *price*) **and** (**least** *mileage*);
In this query, "$A_i = v_{1,1}$ *else* $A_i = v_{1,2}$" means that value $v_{1,1}$ is strictly preferred to value $v_{1,2}$ for attribute A_i. It is assumed that any domain value which is absent

[1] $u \succ v$ means u is preferred to v. $u \succcurlyeq v$ means u is at least as good as v, i.e., $u \succcurlyeq v \Leftrightarrow u \succ v \vee u \approx v$, where $\approx$ denotes indifference.

from a preference clause is less preferred than any value explicitly specified in the clause (but it is not absolutely rejected). Here, the tuples that are not dominated in the sense of the *preferring* clause are $\{t_3, t_4, t_7\}$. Indeed, t_7 dominates t_1, t_2, and t_5, whereas every tuple dominates t_6 except t_2.

Notice that if we add the preference criterion (*color* = 'blue' **else** *color* = 'red' **else** *color* = 'green') to the query, then the skyline is $\{t_1, t_2, t_3, t_4, t_5, t_7\}$, i.e., allmost all of the tuples are incomparable. ◇

3 Different Types of Fuzzy Skylines

There may be many different motivations for making skylines fuzzy in a way or another. First, one may want to refine the skyline by introducing some ordering between its points in order to single out the most interesting ones. Second, one may like to make it more flexible by adding points that strictly speaking do not belong to it, but are close to belonging to it. Third, one may try to simplify the skyline either by granulating the scales of the criteria, or by considering that some criteria are less important than others, or even that some criteria compensate each other, which may enable us to cluster points that are somewhat similar. Fourth, the skyline may be "fuzzy" due to the uncertainty or the imprecision present in the data. Lastly, the preference ordering on some criteria may depend on the context, and may be specified only for some particular or typical contexts. We now briefly review each of these ideas.

3.1 Refining the Skyline

The first idea stated above corresponds to refining S by stating that u is in the fuzzy skyline S_{MP} if i) it belongs to S, ii) $\forall u'$ such that $u >_{dom} u'$, $\exists i$ such that u_i is *much preferred* to u'_i, denoted $(u_i, u'_i) \in MP_i$, which can be expressed:

$$u \in S_{MP} \Leftrightarrow u \in S \wedge \forall u' \in Dm(u),\ \exists i \in \{1, \ldots, n\} \text{ s.t. } (u_i, u'_i) \in MP_i \quad (5)$$

(where $\forall i, (u_i, u'_i) \in MP_i \Rightarrow u_i \succ_i u'_i$; we also assume that MP_i agrees with $\succcurlyeq_i$: $u_i \succcurlyeq_i u'_i$ and $(u'_i, u''_i) \in MP_i \Rightarrow (u_i, u''_i) \in MP_i$). (5) can be equivalently written:

$$\begin{aligned} u \in S_{MP} \Leftrightarrow \forall u' \in D,\ (\neg(u' >_{dom} u) \wedge \\ (u >_{dom} u' \Rightarrow \exists i \in \{1, \ldots, n\} \text{ such that } (u_i, u'_i) \in MP_i)) \end{aligned} \quad (6)$$

Note that u is in S_{MP} if it is incomparable with every other tuple or if it is highly preferred on at least one attribute to every tuple it dominates.

When MP_i becomes gradual, we need to use a fuzzy implication such that $1 \rightarrow q = q$ and $0 \rightarrow q = 1$, which can be expressed as $max(1 - p,\ q)$ (with $p \in \{0, 1\}$). The previous formulas can be translated into fuzzy set terms by:

$$\begin{aligned} \mu_{S_{MP}}(u) = min_{u' \in D}\ min(1 - \mu_{dom}(u',\ u), \\ max(1 - \mu_{dom}(u,\ u'),\ max_i\ \mu_{MP_i}(u_i,\ u'_i))) \end{aligned} \quad (7)$$

where $\mu_{dom}(u, u') = 1$ if u dominates u' and is 0 otherwise, and $\mu_{MP_i}(u_i, u'_i)$ is the extent to which u_i is much preferred to u'_i (where $\mu_{MP_i}(u_i, u'_i) > 0 \Rightarrow u_i \succ_i u'_i$).

Moreover, we assume $u_i \succcurlyeq_i u_i' \Rightarrow \mu_{MP_i}(u_i, u_i'') \geq \mu_{MP_i}(u_i', u_i'')$. Clearly, we have $S_{MP} = S$ when MP_i reduces to the crisp relation $\succ_i$. S_{MP} may of course be non normalized (no tuple gets degree 1), or even empty.

Example 2. Let us consider again the data and query from Example 1 (without the criterion on *color*). Let us introduce the relations $\mu_{\ll_{price}}(x, y) = 1$ if $y - x \geq 1000$, 0 otherwise, and $\mu_{\ll_{mileage}}(x, y) = 1$ if $y - x \geq 5000$, 0 otherwise. Let us assume that the notion "much preferred" is defined as $\forall(x, y)$, $\mu_{MP}(x, y) = 0$ for attributes *make* and *category*. The result is now the set $\{t_3, t_4\}$. Tuple t_7 does not belong to the skyline anymore since at least one of the tuples that it dominates (here t_5) is not *highly* dominated by it.

Notice that if the *mileage* value in tuple t_7 were changed into 20,000, tuple t_7 would then belong to S_{MP}. This situation occurs when the "weakening" of a tuple makes it incomparable with all the others.$\diamond$

A still more refined fuzzy skyline S^* selects those tuples u from S_{MP}, if any, that are such that $\forall u' \in Inc(u)$, $\nexists j$ such that u_j' is much preferred to u_j:

$$u \in S^* \Leftrightarrow u \in S_{MP} \wedge \\ \forall u' \in Inc(u),\ \nexists j \in \{1, \ldots, n\} \text{ such that } (u_j',\ u_j) \in MP_i. \tag{8}$$

Thus, the graded counterpart of Formula (8) is:

$$\mu_{S^*}(u) = min(\mu_{S_{MP}}(u),\ min_{u' \in Inc(u)}\ (1 - max_j\ \mu_{MP_j}(u_j', u_j)) \tag{9}$$

S^* gathers the most interesting points, since they are much better on at least one attribute than the tuples they dominate, and not so bad on the other attributes (w.r.t. other non comparable points).

Example 3. Using the same data and strengthened preferences as in Example 2, we get $S^* = \emptyset$ since both t_3 and t_4 are much worse than t_2 (and t_7) on *price*. $\diamond$

S^* and S_{MP} do not seem to have been previously considered in the literature.

3.2 Making the Skyline More Flexible

Rather than refining the skyline, a second type of fuzzy skyline (denoted by S_{REL} hereafter) corresponds to the idea of relaxing it, i.e., u still belongs to the skyline to some extent (but to a less extent), if u is only weakly dominated by any other u'. Then, $u \in S_{REL}$ iff it is false that there exists a tuple u' much preferred to u w.r.t. all attributes (this expression was proposed in [9]). Formally, one has:

$$u \in S_{REL} \Leftrightarrow \nexists u' \in D,\ \forall i \in \{1, \ldots, n\},\ (u_i',\ u_i) \in MP_i \tag{10}$$

or in fuzzy set terms:

$$\begin{aligned} \mu_{S_{REL}}(u) &= 1 - max_{u' \in D}\ min_i\ \mu_{MP_i}(u_i',\ u_i) \\ &= min_{u' \in D}\ max_i\ 1 - \mu_{MP_i}(u_i',\ u_i). \end{aligned} \tag{11}$$

Example 4. Let us use the same data and query as in Example 1, and assume that "much preferred" is defined as "preferred" on attributes *make* and *category*, and as in Example 2 for attributes *price* and *mileage*. We get $S_{REL} = \{t_1, t_2, t_3, t_4, t_5, t_7\}$ instead of $S = \{t_3, t_4, t_7\}$ since neither t_1, t_2, nor t_5 is highly dominated by any other tuple on all the attributes.$\diamond$

One has $S \subseteq S_{REL}$, i.e., $\forall u \in D$, $\mu_S(u) \leq \mu_{S_{REL}}(u)$, since Formula (2) writes

$$\begin{aligned}\mu_S(u) &= min_{u' \in D}\, 1 - \mu_{dom}(u',\, u) \\ &= min_{u' \in D}\, 1 - min(min_i\, \mu_{\succcurlyeq_i}(u'_i,\, u_i),\, max_i\, \mu_{\succ_i}(u'_i,\, u_i)) \\ &= min_{u' \in D}\, max(max_i\, \mu_{\prec_i}(u'_i,\, u_i),\, min_i\, \mu_{\preccurlyeq_i}(u'_i,\, u_i))\end{aligned}$$

in fuzzy set terms and $max(max_i \mu_{\prec_i}(u'_i, u_i), min_i \mu_{\preccurlyeq_i}(u'_i, u_i)) \leq max_i \mu_{\preccurlyeq_i}(u'_i, u_i) \leq max_i\, 1 - \mu_{MP_i}(u'_i,\, u_i)$ since $1 - \mu_{\preccurlyeq_i}(u'_i,\, u_i) = \mu_{\succ_i}(u'_i,\, u_i) \leq \mu_{MP_i}(u'_i,\, u_i)$. So, one finally has:

$$S^* \subseteq S_{MP} \subseteq S \subseteq S_{REL}. \tag{12}$$

Note that Formula (10) may seem very permissive, but in case we would think of replacing $\forall i$ by $\exists i$, we would lose $S \subseteq S_{REL}$, which is in contradiction with the idea of enlarging S.

Another way of relaxing S is to consider that u still belongs to a fuzzily extended skyline S_{FE} if u is close to u' with $u' \in S$. This leads to the following definition

$$u \in S_{FE} \Leftrightarrow \exists u' \in S \text{ such that } \forall i,\, (u_i,\, u'_i) \in E_i \tag{13}$$

where E_i is a reflexive, symmetrical approximate indifference (or equality) relation defined on the domain of A_i, such that $(u_i,\, u''_i) \in E_i$ and $u_i \preccurlyeq_i u'_i \preccurlyeq_i u''_i \Rightarrow (u_i,\, u'_i) \in E_i$. Moreover, it is natural to assume that $(u_i,\, u'_i) \in E_i \Rightarrow (u_i,\, u'_i) \notin MP_i$ and $(u'_i,\, u_i) \notin MP_i$. S_{FE} can be expressed in fuzzy set terms (then assuming $\mu_{E_i}(u_i,\, u'_i) > 0 \Rightarrow \mu_{MP_i}(u_i,\, u'_i) = 0$, i.e. in other words, $\underline{support}(E_i) = \{(u_i,\, u'_i) | \mu_{E_i}(u_i,\, u'_i) > 0\} \subseteq \{(u_i,\, u'_i) | 1 - \mu_{MP_i}(u_i,\, u'_i) = 1\} = core(\overline{MP_i})$). We also assume $u_i \preccurlyeq_i u'_i \preccurlyeq_i u''_i \Rightarrow \mu_{E_i}(u_i,\, u'_i) \geq \mu_{E_i}(u_i,\, u''_i)$. We have

$$\mu_{S_{FE}}(u) = max_{u' \in D}\, min(\mu_S(u'),\, min_i\, \mu_{E_i}(u_i,\, u'_i)) \tag{14}$$

Then we have the following inclusions.

$$S \subseteq S_{FE} \subseteq S_{REL} \tag{15}$$

Proof. $S \subseteq S_{FE}$. Clearly, $\mu_S \leq \mu_{S_{FE}}$, since the approximate equality relations E_i are reflexive (i.e.,$\forall i, \forall u_i, \mu_{E_i}(u_i,\, u_i) = 1$).

$S_{FE} \subseteq S_{REL}$. Let us show it in the non fuzzy case first, by establishing that the assumption $u \notin S_{REL}$ and $u \in S_{FE}$ leads to a contradiction. Since $u \notin S_{REL}$, $\exists \mathring{u} \in D$ s.t. $\forall i, (\mathring{u}_i, u_i) \in MP_i$. Besides, since $u \in S_{FE}$, $\exists u^* \in S, \forall i, (u^*_i, u_i) \in E_i$. Observe that u^* does not dominate $\mathring{u}$ (since $\forall i, u^*_i \succcurlyeq_i \mathring{u}_i$ entails $(u^*_i,\, u_i) \in MP_i$, due to $\forall i, (\mathring{u}_i,\, u_i) \in MP_i$, which contradicts $\forall i, (u^*_i,\, u_i) \in E_i$). $\mathring{u}$ does not dominate u^* either (since $u^* \in S$). Then, $\mathring{u}$ and u^* are incomparable, and $\exists j, \exists k, u^*_j \succcurlyeq_j \mathring{u}_j$ and $u^*_k \prec_k \mathring{u}_k$. But, we know that in particular $(\mathring{u}_j,\, u_j) \in$

MP_j and $(u_j^*, u_j) \in E_j$. Let us show by *reductio ad absurdum* that it entails $\mathring{u}_j \succ_j u_j^*$. Indeed, assume $\mathring{u}_j \preccurlyeq_j u_j^*$; $(\mathring{u}_j, u_j) \in MP_j$ entails $\mathring{u}_j \succcurlyeq_j u_j$ and $(\mathring{u}_j, u_j) \notin E_j$. Hence a contradiction since $u_j^* \succcurlyeq_j \mathring{u}_j \succcurlyeq_j u_j$ and $(u_j^*, u_j) \in E_j$ (w.r.t. a property assumed for E_j). But in turn, $\mathring{u}_j \succ_j u_j^*$ contradicts that $\exists j, u_j^* \succcurlyeq_j \mathring{u}_j$, the hypothesis we start with. Thus, assuming $u \notin S_{REL}$ leads to a contradiction. The fuzzy case can be handled similarly by working with the cores and supports of fuzzy relations. ∎

3.3 Simplifying the Skyline

On the contrary, it may be desirable to simplify the skyline, for example because it contains too many points. There are many ways to do it. The definitions of S^* and S_{MP} serve this purpose, but they may be empty as already said. We now briefly mention three other meaningful ways to simplify the skyline.

Simplification Through Criteria Weighting. First, one may consider that the set of criteria is partitioned into subsets of decreasing importance, denoted by $W_1, \ldots, W_k$ (where W_1 gathers the criteria of maximal importance). Then we may judge that a tuple cannot belong to the skyline only because it strictly dominates all the other tuples on a non fully important criterion. Indeed it may look strange that a tuple belongs to the skyline while it is dominated on all the important criteria, even if its value on a secondary criterion makes the tuple finally incomparable. In this view, less important criteria may be only used to get rid of tuples that are dominated on immediately less important criteria, in case of ties on more important criteria. Let us introduce the following definitions underlying the concept of a hierarchical skyline.

$$u >_{dom_{W_i}} u' \Leftrightarrow \forall j \text{ such that } c_j \in W_i, \\ (u_j \succcurlyeq_j u'_j \wedge \exists p \text{ such that } (c_p \in W_i \wedge u_p \succ_p u'_p)). \tag{16}$$

$$\forall i \in \{1, \ldots, k\}, u \in S_{W_i} \Leftrightarrow u \in S_{W_{i-1}} \wedge \forall u' \in D, \neg(u >_{dom_{W_i}} u') \tag{17}$$

assuming $\forall u \in D$, $u \in S_{W_0}$. The set S_{W_j} gathers the tuples that are not dominated by any other in the sense of the criteria in $W_1 \cup ... \cup W_j$. By construction, one has:

$$S_{W_1} \supseteq S_{W_2} \supseteq \ldots \supseteq S_{W_k}.$$

In the same spirit, in [12], an operator called *cascade* iteratively eliminates the dominated tuples in each level of a preference hierarchy. Prioritized composition of preferences obeying the same concept can also be modeled by the operator *winnow* proposed by Chomicki [11].

Example 5. Let us consider the data from Table 1, and the query:

select * **from** car **preferring**
((category = 'sedan' **else** category = 'roadster' **else** category = 'coupe') **and**
(color = 'blue' **else** color = 'red' **else** color = 'green')) (W_1)

cascade (**least** price) (W_2);

We get the nested results: $S_{W_1} = \{t_1, t_7\}$ and $S_{W_2} = \{t_7\}$. $\diamond$

An alternative solution — which does not make use of priorities but is rather based on counting — is proposed in [3,4] where the authors introduce a concept called *k-dominant skyline*, which relaxes the idea of dominance to k-dominance. A point p is said to k-dominate another point q if there are k ($\leq d$) dimensions in which p is better than or equal to q and is better in at least one of these k dimensions. A point that is not k-dominated by any other points is in the k-dominant skyline. Still another method for defining an order for two incomparable tuples is proposed in [5], based on the number of other tuples that each of the two tuples dominates (notion of k-representative dominance).

Simplification Through The Use of Coarser Scales. A second, completely different idea for simplifying a skyline is to use coarser scales for the evaluation of the attributes (e.g., moving from precise values to rounded values). This may lead to more comparable (or even identical) tuples. Notice that the skyline obtained after simplification does not necessarily contain less points than the initial one (cf. the example hereafter). However, the tuples that *become* member of the skyline after modifying the scale are in fact *equivalent* preferencewise.

Example 6. Let us consider a relation r of schema (A, B) containing the tuples $t_1 = \langle 15.1, 7\rangle$, $t_2 = \langle 15.2, 6\rangle$, and $t_3 = \langle 15.3, 5\rangle$, and the skyline query looking for those tuples which have the smallest value for both attributes A and B. Initially, the skyline consists of all three tuples t_1, t_2, t_3 since none of them is dominated by another. Using rounded values for evaluating A and B one gets $\{t_3\}$ as the new skyline. Let us now consider that relation r contains the tuples $t'_1 = \langle 15.1, 5.1\rangle$, $t'_2 = \langle 15.2, 5.2\rangle$, and $t'_3 = \langle 15.3, 5.4\rangle$. This time, the initial skyline is made of the sole tuple t'_1 whereas the skyline obtained by simplifying the scales is $\{t'_1, t'_2, t'_3\}$. $\diamond$

Simplification Through the Use of k-discrimin. Still another way to increase the number of comparable tuples is to use a *2-discrimin* (or more generally an order *k-discrimin*) ordering (see [13] from which most of the following presentation is drawn). A definition of classical *discrimin* relies on the set of criteria not respected in the same way by both tuples u and v, denoted by $D_1(u, v)$ [14]:

$$D_1(u, v) = \{c_i \in C \mid v_i = u_i\} \tag{18}$$

$$u >_{disc} v \Leftrightarrow min_{c_i \notin D_1(u,\, v)}\, u_i > min_{c_i \notin D(u,\, v)}\, v_i \tag{19}$$

Discrimin-optimal solutions are also Pareto-optimal but not conversely, in general (see [14]).

Classical discrimin is based on the elimination of identical singletons at the same places in the comparison process of the two sequences. Thus with classical

discrimin, comparing $u = (0.2, 0.5, 0.3, 0.4, 0.8)$ and $v = (0.2, 0.3, 0.5, 0.6, 0.8)$ amounts to comparing vectors u' and v' where $u' = (0.5, 0.3, 0.4)$ and $v' = (0.3, 0.5, 0.6)$ since $u_1 = v_1 = 0.2$ and $u_5 = v_5 = 0.8$. Thus, $u =_{min} v$ and we still have $u =_{discrimin} v$. More generally, we can work with 2-element subsets which are identical and pertain to the same pair of criteria. Namely in the above example, we may consider that (0.5, 0.3) and (0.3, 0.5) are "equilibrating" each other. Note that it supposes that the two corresponding criteria have the same importance. Then we delete them, and we are led to compare $u'' = (0.4)$ and $v'' = (0.6)$. Let us take another example: $u_2 = (0.5, 0.4, 0.3, 0.7, 0.9)$ and $v_2 = (0.3, 0.9, 0.5, 0.4, 1)$. Then, we would again delete (0.5, 0.3) with (0.3, 0.5) yielding $u'_2 = (0.4, 0.7, 0.9)$ and $v'_2 = (0.9, 0.4, 1)$. Note that in this example we do not simplify 0.4, 0.9 with 0.9, 0.4 since they do not pertain to the same pair of criteria. Note also that simplifications can take place only one time. Thus, if the vectors are of the form $u = (x, y, x, s)$ and $v = (y, x, y, t)$ (with $min(x, y) \leq min(s, t)$ in order to have the two vectors min-equivalent), we may either delete components of ranks 1 and 2, or of ranks 2 and 3, leading in both cases to compare (x, s) and (y, t), and to consider the first vector as smaller in the sense of the order 2-discrimin, as soon as $x < min(y, s, t)$.

We can now introduce the definition of the (order) 2-discrimin [13]. Let us build a set $D_2(u, v)$ as $\{(c_i, c_j) \in C \times C$, such that $u_i = v_j$ and $u_j = v_i$ and if there are several such pairs, they have no common components$\}$. Then the 2-discrimin is just the minimum-based ordering once components corresponding to pairs in $D_2(u, v)$ and singletons in $D_1(u, v)$ are deleted. Note that $D_2(u, v)$ is not always unique as shown by the above example. However this does not affect the result of the comparison of the vectors after the deletion of the components as it can be checked from the above formal example, since the minimum aggregation is not sensitive to the place of the terms. Notice that the k-discrimin requires stronger assumptions than Pareto-ordering since it assumes that the values related to different attributes are comparable (which is the case for instance when these values are obtained through scoring functions).

This idea of using *k-discrimin* for simplifying a skyline can be illustrated by the following example, where we compare hotels on the basis of their price, distance to the station, and distance to a conference location (which should all be minimized). Then (80, 1, 3) et (70, 3, 1) are not Pareto comparable, while we may consider that the two distance criteria play similar roles and that there is equivalence between the sub-tuples (1, 3) and (3, 1) leading to compare the tuples on the remaining components.

3.4 Dealing with Uncertain Data

The fourth type of "fuzzy" skyline is quite clear. When attributes values are imprecisely or more generally fuzzily known, we are led to define the tuples that certainly belong to the skyline, and those that only possibly belong to it, using necessity and possibility measures. This idea was suggested in [8].

3.5 Dealing with Incomplete Contextual Preferences

In [15], we concentrate on the last category of "fuzzy" skyline that is induced by an incompletely known context-dependency of the involved preferences. In order to illustrate this, let us use an example taken from [16], which consists of a relation with three attributes *Price*, *Distance* and *Amenity* about a set of hotels (see Table 2). A skyline query may search for those hotels for which there is *no cheaper* and, at the same time, *closer* to the beach alternative. One can easily check that the skyline contains hotels h_4 and h_5. In other terms, hotels h_4 and h_5 represent non-dominated hotels w.r.t. *Price* and *Distance* dimensions.

Table 2. Relation describing hotels

Hotel	Price	Distance	Amenity
h_1	200	10	Pool(P)
h_2	300	10	Spa(S)
h_3	400	15	Internet'I)
h_4	200	5	Gym(G)
h_5	100	20	Internet(I)

Table 3. Contextual Skylines

Context	Preferences	Skyline
C_1: Business, June	$I \succ G, I \succ \{P, S\}, G \succ \{P, S\}$	h_3, h_4, h_5
C_2: Vacation	$S \succ \{P, I, G\}$	h_2, h_4, h_5
C_3: Summer	$P \succ \{I, G\}$ $S \succ \{I, G\}$	h_1, h_2, h_4, h_5
C_q: Business, Summer	–	?

Let us now assume that the preferences on attribute *Amenity* depend on the *context*. For instance, let us consider the three contexts C_1, C_2 and C_3 shown in Table 3 (where a given context can be composed at most by two context parameters (*Purpose*, *Period*)). For example, when the user is on a business trip in June (context C_1), hotels h_3, h_4 and h_5 are the results of the skyline query for C_1. See Table 3 for contexts C_2 and C_3 and their corresponding skylines.

Let us now examine situation C_q (fourth row in Table 3), where the user plans a business trip in the summer but states no preferences. Considering amenities *Internet* (I) and *Pool* (P), one can observe that: (i) I may be preferred to P as in C_1; (ii) P may be preferred to I as in C_3, or (iii) I and P may be equally favorable as in C_2. Moreover, the uncertainty propagates to the dominance relationships, i.e., every hotel may dominate another with a certainty degree that depends on the context. In [15], it is shown how a set of plausible preferences suitable for the context at hand may be derived, on the basis of the information known for other contexts (using a CBR-like approach). Uncertain dominance relationships

are modeled in the setting of possibility theory. In this framework, the user is provided with the tuples that are not dominated with a high certainty, leading to a notion of possibilistic contextual skyline. It is also suggested how possibilistic logic can be used to handle contexts with conflicting preferences, as well as dependencies between contexts.

4 Conclusion

The paper has provided a structured discussion of different types of "fuzzy" skylines. Five lines of extension have been considered. First, one has refined he skyline by introducing some ordering between its points in order to single out the most interesting ones. Second, one has made it more flexible by adding points that strictly speaking do not belong to it, but are close to belonging to it. Third, one has aimed at simplifying the skyline either by granulating the scales of the criteria, or by considering that some criteria are less important than others, or even that some criteria compensate each other. Fourth, the case where the skyline is fuzzy due to the uncertainty in the data has been dealt with. Lastly, skyline queries has been generalized to incompletely stated context-dependent preferences.

Among perspectives for future research, let us mention: (i) the integration of these constructs into a database language based on SQL, (ii) the study of query optimization aspects. In particular, it would be worth investigating whether some techniques proposed in the context of Skyline queries on classical data (for instance those based on presorting, see, e.g., [17]) could be adapted to (some of) the fuzzy skyline queries discussed here.

References

1. Hadjali, A., Kaci, S., Prade, H.: Database preferences queries: a possibilistic logic approach with symbolic priorities. In: Proc. FoIKS, pp. 291–310 (2008)
2. Borzsony, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proc. ICDE, pp. 421–430 (2001)
3. Chan, C.-Y., Jagadish, H.V., Tan, K.-L., Tung, A.K.H., Zhang, Z.: Finding k-dominant skylines in high dimensional space. In: ACM SIGMOD 2006, pp. 503–514 (2006)
4. Chan, C.Y., Jagadish, H.V., Tan, K.L., Tung, A.K.H., Zhang, Z.: On high dimensional skylines. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 478–495. Springer, Heidelberg (2006)
5. Lin, X., Yuan, Y., Zhang, Q., Zhang, Y.: Selecting stars: The k most representative skyline operator. In: Proc. ICDE, 2007, pp. 86–95 (2007)
6. Khalefa, M.E., Mokbel, M.F., Levandoski, J.J.: Skyline query processing for incomplete data. In: Proc. ICDE, pp. 556–565 (2008)
7. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic skylines on uncertain data. In: VLDB 2007, pp. 15–26 (2007)

8. Hüllermeier, E., Vladimirskiy, I., Prados Suárez, B., Stauch, E.: Supporting case-based retrieval by similarity skylines: Basic concepts and extensions. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 240–254. Springer, Heidelberg (2008)
9. Goncalves, M., Tineo, L.J.: Fuzzy dominance skyline queries. In: Wagner, R., Revell, N., Pernul, G. (eds.) DEXA 2007. LNCS, vol. 4653, pp. 469–478. Springer, Heidelberg (2007)
10. Zadrozny, S., Kacprzyk, J.: Bipolar queries and queries with preferences. In: Proc. of FlexDBIST 2006, pp. 415–419 (2006)
11. Chomicki, J.: Preference formulas in relational queries. ACM Transactions on Database Systems 28(4), 427–466 (2003)
12. Kießling, W., Köstler, G.: Preference SQL — design, implementation, experiences. In: Proc. of the 2002 VLDB Conference, pp. 990–1001 (2002)
13. Prade, H.: Refinement of Minimum-Based Ordering in between Discrimin and Leximin. In: Proc. Linz Seminar on Fuzzy Set Theory, pp. 39–43 (2001)
14. Dubois, D., Fargier, H., Prade, H.: Fuzzy constraints in job-shop scheduling. Journal of Intelligent Manufacturing 6, 215–234 (1995)
15. Hadjali, A., Pivert, O., Prade, H.: Possibilistic contextual skylines with incomplete preferences. In: Proc. of the 2nd IEEE International Conference on Soft Computing and Pattern Recognition (SoCPaR 2010), Cergy-Pontoise, France (2010)
16. Sacharidis, D., Arvanitis, A., Sellis, T.: Probabilistic contextual skylines. In: ICDE, pp. 273–284 (2010)
17. Bartolini, I., Ciaccia, P., Patella, M.: Efficient sort-based skyline evaluation. ACM Trans. Database Syst. 33(4), 1–49 (2008)

On Database Queries Involving Inferred Fuzzy Predicates

Olivier Pivert, Allel Hadjali, and Grégory Smits

Technopole Anticipa 22305 Lannion Cedex France
Irisa – Enssat, IUT Lannion
Technopole Anticipa 22305 Lannion Cedex France
{pivert,hadjali}@enssat.fr, gregory.smits@univ-rennes1.fr

Abstract. This paper deals with database preference queries involving fuzzy conditions which do not explicitly refer to an attribute from the database, but whose meaning is rather inferred from a set of rules. The approach we propose, which is based on some concepts from the fuzzy control domain (aggregation and defuzzification, in particular), significantly increases the expressivity of fuzzy query languages inasmuch as it allows for new types of predicates. An implementation strategy involving a coupling between a DBMS and a fuzzy reasoner is outlined.

1 Introduction

In database research, the last decade has witnessed a growing interest in preference queries. Motivations for introducing preferences inside database queries are manifold [1]. First, it has appeared to be desirable to offer more expressive query languages that can be more faithful to what a user intends to say. Second, the introduction of preferences in queries provides a basis for rank-ordering the retrieved items, which is especially valuable in case of large sets of items satisfying a query. Third, a classical query may also have an empty set of answers, while a relaxed (and thus less restrictive) version of the query might be matched by items in the database.

Approaches to database preference queries may be classified into two categories according to their qualitative or quantitative nature [1]. In the latter, preferences are expressed quantitatively by a monotone scoring function, and the overall score is positively correlated with partial scores. Since the scoring function associates each tuple with a numerical score, tuple t_1 is preferred to tuple t_2 if the score of t_1 is higher than the score of t_2. Representatives of this family of approaches are top-k queries [2] and fuzzy-set-based approaches (e.g., [3]). In the qualitative approach, preferences are defined through binary preference relations. Representatives of qualitative approaches are those relying on a dominance relationship, e.g. Pareto order, in particular *Preference SQL* [4] and Skyline queries [5] and the approach presented in [6].

In this paper, we focus on the fuzzy-set-based approach to preference queries, which is founded on the use of fuzzy set membership functions that describe the

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 592–601, 2011.

preference profiles of the user on each attribute domain involved in the query. The framework considered is that of the fuzzy query language called SQLf [3].

The objective is to extend SQLf so as to authorize the use, inside queries, of fuzzy predicates for which *there does not exist any underlying attribute* in the database [7]. In such a case, it is of course impossible to explicitly define the membership function attached to the fuzzy predicate. It is rather assumed that the satisfaction level of such a predicate P depends on the satisfaction level of other predicates $C_1, \ldots, C_n$, but it is in general not easy to express its membership function μ_P as a simple aggregation of the μ_{C_i}'s. A solution is to use fuzzy rules, somewhat in the spirit of [8] where fuzzy preferences are inferred from the fuzzy context of the user (but the technique is different since one needs to derive satisfaction degrees, not membership functions). A first approach to the modelling of inferred fuzzy predicates has been defined in [7], and we will first point out the shortcomings of this model. Then, we will propose an alternative approach based on an inference technique used in fuzzy controllers.

The remainder of the paper is structured as follows. Section 2 consists of a short reminder about fuzzy sets and fuzzy queries. Section 3 provides a critical discussion of the approach proposed in [7] for modelling inferred fuzzy predicates. In Section 4, we present an alternative approach, based on fuzzy rules and some concepts from the fuzzy control field. Implementation aspects are dealt with in Section 5, whereas Section 6 summarizes the contributions and outlines some perspectives for future work.

2 Reminder about Fuzzy Sets and Fuzzy Queries

2.1 Basic Notions about Fuzzy Sets

Fuzzy set theory was introduced by Zadeh [9] for modeling classes or sets whose boundaries are not clear-cut. For such objects, the transition between full membership and full mismatch is gradual rather than crisp. Typical examples of such fuzzy classes are those described using adjectives of the natural language, such as *young, cheap, fast*, etc. Formally, a fuzzy set F on a referential U is characterized by a membership function $\mu_F : U \rightarrow [0, 1]$ where $\mu_F(u)$ denotes the grade of membership of u in F. In particular, $\mu_F(u) = 1$ reflects full membership of u in F, while $\mu_F(u) = 0$ expresses absolute non-membership. When $0 < \mu_F(u) < 1$, one speaks of partial membership.

Two crisp sets are of particular interest when defining a fuzzy set F:

- the core $C(F) = \{u \in U \mid \mu_F(u) = 1\}$, which gathers the *prototypes* of F,
- the support $S(F) = \{u \in U \mid \mu_F(u) > 0\}$.

In practice, the membership function associated with F is often of a trapezoidal shape. Then, F is expressed by the quadruplet (A, B, a, b) where $C(F) = [A, B]$ and $S(F) = [A - a, B + b]$, see Figure 1.

Let F and G be two fuzzy sets on the universe U, we say that $F \subseteq G$ iff $\mu_F(u) \leq \mu_G(u)$, $\forall u \in U$. The complement of F, denoted by F^c, is defined by

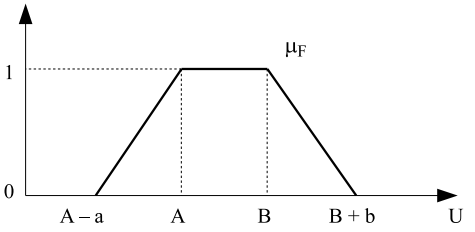


Fig. 1. Trapezoidal membership function

$\mu_{F^c}(u) = 1 - \mu_F(u)$. Furthermore, $F \cap G$ (resp. $F \cup G$) is defined the following way: $\mu_{F \cap G} = min(\mu_F(u), \mu_G(u))$ (resp. $\mu_{F \cup G} = max(\mu_F(u), \mu_G(u))$).

As usual, the logical counterparts of the theoretical set operators $\cap$, $\cup$ and complementation operator correspond respectively to the conjunction $\wedge$, disjunction $\vee$ and negation $\neg$. See [10] for more details.

2.2 About SQLf

The language called SQLf described in [3] extends SQL so as to support fuzzy queries. The general principle consists in introducing gradual predicates wherever it makes sense. The three clauses *select*, *from* and *where* of the base block of SQL are kept in SQLf and the "from" clause remains unchanged. The principal differences affect mainly two aspects :

- the calibration of the result since it is made with discriminated elements, which can be achieved through a number of desired answers (k), a minimal level of satisfaction (α), or both, and
- the nature of the authorized conditions as mentioned previously.

Therefore, the base block is expressed as:

select [**distinct**] [$k \mid \alpha \mid k$, α] attributes **from** relations **where** fuzzy-cond

where "fuzzy-cond" may involve both Boolean and fuzzy predicates. This expression is interpreted as:

- the fuzzy selection of the Cartesian product of the relations appearing in the *from* clause,
- a projection over the attributes of the *select* clause (duplicates are kept by default, and if *distinct* is specified the maximal degree is attached to the representative in the result),
- the calibration of the result (top k elements and/or those whose score is over the threshold α).

The operations from the relational algebra — on which SQLf is based — are extended to fuzzy relations by considering fuzzy relations as fuzzy sets on the one hand and by introducing gradual predicates in the appropriate operations

(selections and joins especially) on the other hand. The definitions of these extended relational operators can be found in [11]. As an illustration, we give the definitions of the fuzzy selection and join operators hereafter, where r and s denote two fuzzy relations defined respectively on the sets of domains X and Y.

- $\mu_{select(r,\, cond)}(t) = \top(\mu_r(t),\, \mu_{cond}(t))$ where *cond* is a fuzzy predicate and $\top$ is a triangular norm (most usually, *min* is used),
- $\mu_{join(r,\, s,\, A,\, B,\, \theta)}(tu) = \top(\mu_r(t),\, \mu_s(u),\, \mu_\theta(t.A,\, u.B))$ where A (resp. B) is a subset of X (resp. Y), A and B are defined over the same domains, θ is a binary relational operator (possibly fuzzy), $t.A$ (resp. $u.B$) stands for the value of t over A (resp. u over B).

3 Inferred Fuzzy Predicates: A First Approach

The situation considered is that where a user wants to express a fuzzy selection condition in his/her query but i) there does not exist any associated attribute in the database whose domain could be used as the referential underlying a fuzzy membership function, ii) it is not possible to express in a simple way the fuzzy condition as an aggregation of elementary fuzzy requirements on different attributes. We first present the approach presented in [7] and point out some of its shortcomings.

3.1 Presentation of the Approach by Koyuncu

An example given in [7] considers a relation *Match* describing soccer matches, with schema (*#id*, *goalPositions*, *goals*, *fouls*, *penalties*, *disqualifications*, *year*), and queries such as: "find the matches played in 2010 with a high harshness level". In this query the fuzzy condition "high harshness level" does not refer to any specific attribute from the relation. The author proposes to give it a semantics by means of rules such as:

if ((*fouls* **is** *several*) **or** (*fouls* **is** *many*)) (with threshold 0.6)
and (*penalties* **is** *average*) (with threshold 0.7)
and (*disqualifications* **is** *average*) (with threshold 0.5)
then *harshness* **is** *high* (with $\mu = Y$).

In this rule (let us denote it by R_1), Y denotes the membership degree attached to the conclusion, and it is computed using i) the degrees attached to the predicates in the left-hand side of the rule, ii) the so-called "matching degree of the rule conclusion", and iii) an implication function (Gödel's implication defined as

$$p \rightarrow_{G\ddot{o}} q = \begin{cases} 1 \text{ if } q \geq p \\ q \text{ otherwise.} \end{cases}$$

is used by the author). Conjunction and disjunction in the left-hand side are interpreted by *min* and *max* respectively.

The "matching degree of the rule conclusion" expresses the extent to which the fuzzy term present in the right-hand side of the rule (*high* in the rule above) corresponds to the fuzzy term involved in the user query (for instance, the user might aim to retrieve matches with a *medium* level of harshness, in which case, the matching degree would assess the similarity between *high* and *medium*).

Example 1. Let us consider the query:

select $\#id$ **from** *Match* **where** *year* $= 2010$ **and** *harshness_level* **is** *medium*

and the following tuple from relation *Match*: $\langle 1,\ 19,\ 8,\ 23,\ 5,\ 3,\ 2010\rangle$. Let us assume that

$$\mu_{several}(23) = 0,\ \mu_{many}(23) = 1,\ \mu_{average}(5) = 1,\ \mu_{average}(3) = 0.5,$$

and $\mu_{sim}(high, medium) = 0.5$.

Since $max(0,\ 1) \geq 0.6$, $1 \geq 0.7$, and $0.5 \geq 0.5$, rule R_1 can be fired. The final truth degree obtained for the predicate "harshness is medium" is equal to:

$$min(max(0,\ 1),\ 1,\ 0.5) \rightarrow_{G\ddot{o}} \mu_{sim}(high, medium) = 0.5 \rightarrow_{G\ddot{o}} 0.5 = 1.\diamond$$

3.2 Critical Discussion

The technique advocated by Koyuncu calls for several comments:

- First and foremost, this approach does not actually infer fuzzy predicates. The truth degree it produces is associated with a gradual *rule* (it is an *implication degree*), not with the fuzzy *term* present in the conclusion of the rule. This way of doing does not correspond to a well-founded inference scheme such as, for instance, the generalized modus ponens [12].
- The use of a similarity relation between linguistic labels for assessing the rule is debatable: on which semantic basis are the similarity degrees defined and by whom? In the context of a logic-based fuzzy inference system, it would make more sense to consider a *compatibility degree* such as that defined in [12], but this would still not solve the problem evoked in the preceding point.
- The use of *local thresholds* in the left-hand sides of the rules is somewhat contradictory with the "fuzzy set philosophy" which is rather oriented toward expressing trade-offs between different gradual conditions.
- It is not clear what happens when several rules involving the same attribute in their conclusions can be fired at the same time.

4 A Fuzzy-Reasoning-Based Approach

The approach we propose aims at defining the semantics of an inferred fuzzy predicate by means of a set of fuzzy rules, as it is done in classical fuzzy control approaches. An example of the type of queries considered is "find the matches played in 2010 which had a *high harshness level* and had *many* goals scored", where *high harshness level* denotes an inferred fuzzy predicate.

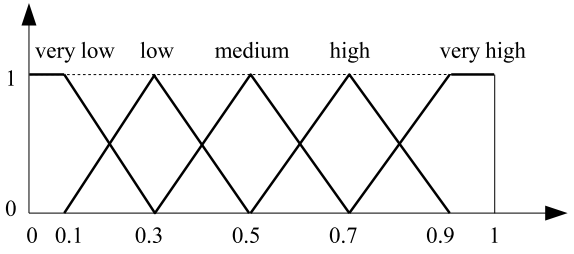


Fig. 2. Fuzzy partition

The principle of the approach we propose is as follows:

- one uses a fuzzy partition over the unit interval, associated with a list of linguistic labels (*very low*, *low*, *medium*, and so on), see an example of such a partition in Figure 2. These labels will be used in the conclusion parts of the fuzzy rules and constitute the basis of the evaluation of the satisfaction degree (in [0, 1]) of the inferred fuzzy predicates present in the query.
- one considers expert-defined fuzzy rules of the form:

  if ((*fouls* **is** *several*) **or** (*fouls* **is** *many*))
  and (*penalties* **is** *average*) **and** (*disqualifications* **is** *average*)
  then *harshness_level* **is** *high*.

  if ((*fouls* **is** *low*) **or** (*fouls* **is** *very low*))
  and (*penalties* **is** *very low*) **and** (*disqualifications* **is** *very low*)
  then *harshness_level* **is** *very low*.

- one computes the degree (in the unit interval) attached to the condition involving an inferred fuzzy predicate by means of a fuzzy reasoner. This is not exactly a "fuzzy controller" — see e.g. [13] — since there is no actual feedback loop. However, the general inference principle, recalled hereafter, is the same.

4.1 Reminder about Fuzzy Control

The following presentation of the general principle of a fuzzy controller is partly drawn from [14]. Let the starting point be a conventional fuzzy controller for temperature control with the temperature x as input and the actuating variable y as controller output, given by the fuzzy rules

R'_1: **if** x **is** *cold* **then** y **is** *big*
R'_2: **if** x **is** *hot* **then** y **is** *small*.

The fuzzy terms *cold*, *hot*, *big*, and *small* are represented in Figure 3.

With a measured temperature x_k at time $t = t_k$ the two fuzzy rules are activated according to the membership degree of x_k to the fuzzy sets *cold* and *hot*.

The membership degree is equivalent to the value of the respective membership function at $x = x_k$. In the example of Figure 3, x_k belongs more to the set *hot* of the hot temperatures than to the set *cold* of the cold temperatures. Following from that, rule R'_2 is more strongly activated than rule R'_1, and the output fuzzy set *small* of the second rule gets a higher weighted in the overall fuzzy set. This output fuzzy set of the entire controller is the set of all possible output values for the input value x_k, and it is computed by association of all differently activated output fuzzy sets of all rules.

More formally, let us consider a set of rules of the form "**if** x **is** A_i **then** y **is** B_i". In the presence of the fact $x = x_k$, one gets a set of partial conclusions of the form (y **is** B'_i) such that

$$\mu_{B'_i}(y) = min(\mu_{A_i}(x_k),\ \mu_{B_i}(y))$$

and the global fuzzy output (y **is** B) is such that

$$\mu_B(y) = max_i\ \mu_{B'_i}(y).$$

Then, a single value value y_k of this output fuzzy set has to be determined by so-called defuzzification. This value will be the controller response to the input value x_k. The most commonly used defuzzification technique is the center-of-gravity method, that means, the y coordinate of the center of gravity of the fuzzy set is used as defuzzification result (see Figure 3).

Considering the transfer function of the entire controller, it can easily be seen, that for any low temperature, that only belongs to the set cold, the controller output is always the same, because for each low temperature only the output set *big* is activated more or less while no other set is activated, and therefore the defuzzification result is always the y coordinate of the center of the triangle

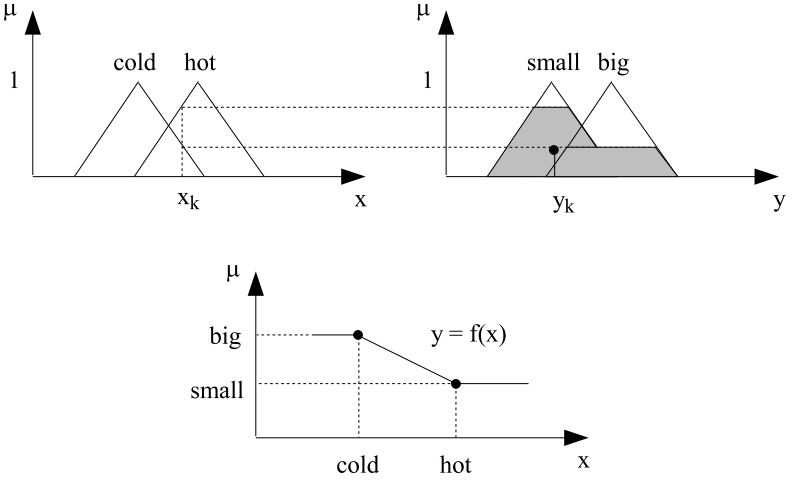


Fig. 3. Principle of a fuzzy controller

big. Analogously, the controller output for high temperatures is always the y coordinate of the center of the triangle *small* . For all other temperatures in between, that belong to the sets *cold* and *hot* more or less as well, the controller output is always between *big* and *small*.

Summarized, the controller transfer function can be defined by a characteristic curve as shown in Figure 3, although the curve between the supporting points is not necessarily linear as in the figure. It depends on the shape of the fuzzy sets and the defuzzification method.

In this example, we have considered only one fuzzy predicate in the antecedent of a rule, but the principle may be straightforwardly generalized to the case where the antecedent is a compound condition involving conjunctions and/or disjunctions. Then, the degree of the antecedent is computed by interpreting *and* and *or* by the triangular norms *minimum* and *maximum* respectively.

4.2 Computation of the Final Degree

In the database query context considered in this paper, the assessment of a tuple t wrt a fuzzy query involving an inferred fuzzy predicate of the form "*H_level is F*" (where F is a fuzzy term) is as follows:

- one fires all of the rules which include the "virtual attribute" *H_level* in their conclusion,
- one aggregates the outputs of these rules and defuzzifies the result so as to obtain the global output value $h \in [0, 1]$,
- one computes the final degree $\mu_F(h)$ using the membership function associated with the fuzzy term F.

5 Implementation Aspects

This approach implies coupling a DBMS with a fuzzy inference engine, according to the architecture sketched in Figure 4.

First, the SQLf query is compiled into a procedural program (called the "evaluator" hereafter) which scans the relation(s) involved. Let us assume that the query involves a global satisfaction threshold α. For each tuple t, the evaluator:

- computes the degrees related to the "regular" fuzzy predicates (i.e., the non-inferred ones) involved in the selection condition,
- sends a request (along with the tuple t itself) to the fuzzy reasoner if necessary (i.e., in the presence of inferred fuzzy predicates in the selection condition). For each inferred predicate φ_i of the form "H_i *is* F_i", the fuzzy reasoner
  - selects the rules which have H_i in their conclusion,
  - computes $\mu_{\varphi_i}(t)$ according to the process described in Subsection 4.2,
  - sends it back to the query evaluator, which
- computes the final degree attached to tuple t,
- discards tuple t if its degree is smaller than α,

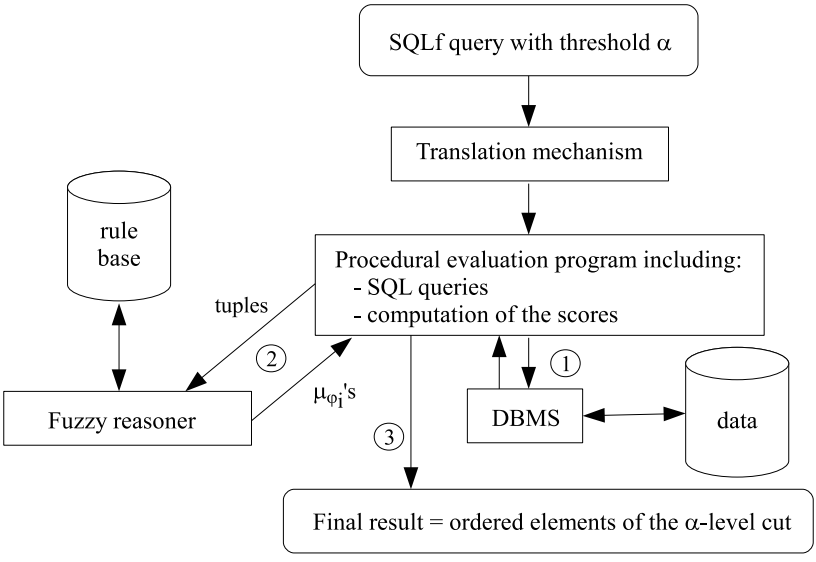


Fig. 4. Query processing strategy

In the case of a conjunctive selection condition involving both "regular" fuzzy predicates and inferred ones, it is possible to use the so-called *derivation method* proposed in [15] so as to avoid an exhaustive scan of the relation(s) concerned (a derived Boolean selection condition is then derived from the part of the initial fuzzy condition which does not include inferred predicates). Indeed, some connections between the fuzzy preference criteria considered and Boolean conditions make it possible to take advantage of the optimization mechanisms offered by classical DBMSs so as to efficiently process fuzzy queries.

6 Conclusion

In this paper, we have proposed an approach to the modelling and handling of *inferred fuzzy predicates*. These are predicates which may be used inside a database preference queries, which do not refer to any attribute from the database, and which cannot be easily defined in terms of a simple aggregation of other atomic predicates. After pointing out the flaws of a previous approach from the literature, we have defined an interpretation model based on a fuzzy rule base and the type of inference used in fuzzy controllers. The way such an inference module may be coupled with a DBMS capable of handling fuzzy queries has been described. An efficient processing technique based on the transformation of (non-inferred) fuzzy predicates into Boolean conditions makes it possible to expect a limited overhead in terms of performances.

As to perspectives for future work, one obviously concerns experimentation. The implementation of a prototype should make it possible to confirm the feasibility of the approach and the fact that reasonable performances may be expected from the evaluation strategy outlined in the present paper. The application of the approach to spatial and temporal databases is also an interesting issue.

References

1. Hadjali, A., Kaci, S., Prade, H.: Database preferences queries – a possibilistic logic approach with symbolic priorities. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 291–310. Springer, Heidelberg (2008)
2. Bruno, N., Chaudhuri, S., Gravano, L.: Top-k selection queries over relational databases: mapping strategies and performance evaluation. ACM Trans. on Database Systems 27, 153–187 (2002)
3. Bosc, P., Pivert, O.: SQLf: a relational database language for fuzzy querying. IEEE Trans. on Fuzzy Systems 3(1), 1–17 (1995)
4. Kießling, W., Köstler, G.: Preference SQL — Design, implementation, experiences. In: Proc. of VLDB 2002, pp. 990–1001 (2002)
5. Bőrzsőnyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proc. of ICDE 2001, pp. 421–430 (2001)
6. Chomicki, J.: Preference formulas in relational queries. ACM Transactions on Database Systems 28(4), 427–466 (2003)
7. Koyuncu, M.: Fuzzy querying in intelligent information systems. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 536–547. Springer, Heidelberg (2009)
8. Hadjali, A., Mokhtari, A., Pivert, O.: A fuzzy-rule-based approach to contextual preference queries. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. LNCS, vol. 6178, pp. 532–541. Springer, Heidelberg (2010)
9. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
10. Dubois, D., Prade, H.: Fundamentals of fuzzy sets. The Handbooks of Fuzzy Sets, vol. 7. Kluwer Academic Pub., Netherlands (2000)
11. Bosc, P., Buckles, B., Petry, F., Pivert, O.: Fuzzy databases. In: Bezdek, J., Dubois, D., Prade, H. (eds.) Fuzzy Sets in Approximate Reasoning and Information Systems. The Handbook of Fuzzy Sets Series, pp. 403–468. Kluwer Academic Publishers, Dordrecht (1999)
12. Dubois, D., Prade, H.: Fuzzy sets in approximate reasoning. Fuzzy Sets and Systems 40(1), 143–202 (1991)
13. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modelling and control. IEEE Transactions on Systems, Man, and Cybernetics 15, 116–132 (1985)
14. Michels, K.: TS control — The link between fuzzy control and classical control theory. In: de Bruin, M., Mache, D., Szabados, J. (eds.) Trends and Applications in Constructive Approximation, pp. 181–194. Birkhäuser Verlag, Basel (2005)
15. Bosc, P., Pivert, O.: SQLf query functionality on top of a regular relational database management system. In: Pons, O., Vila, M., Kacprzyk, J. (eds.) Knowledge Management in Fuzzy Databases, pp. 171–190. Physica-Verlag, Heidelberg (2000)

PMAFC: A New Probabilistic Memetic Algorithm Based Fuzzy Clustering

Indrajit Saha[1], Ujjwal Maulik[2], and Dariusz Plewczynski[1]

[1] Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, 02-106 Warsaw, Poland
{indra,darman}@icm.edu.pl
[2] Department of Computer Science and Engineering, Jadavpur University, Jadavpur-700032, West Bengal, India
umaulik@cse.jdvu.ac.in

Abstract. In this article, a new stochastic approach in form of memetic algorithm for fuzzy clustering is presented. The proposed probabilistic memetic algorithm based fuzzy clustering technique uses real-coded encoding of the cluster centres and two fuzzy clustering validity measures to compute *a priori* probability for an objective function. Moreover, the adaptive arithmetic recombination and opposite based local search techniques are used to get better performance of the proposed algorithm by exploring the search space more powerfully. The performance of the proposed clustering algorithm has been compared with that of some well-known existing clustering algorithms for four synthetic and two real life data sets. Statistical significance test based on analysis of variance (ANOVA) has been conducted to establish the statistical significance of the superior performance of the proposed clustering algorithm. Matlab version of the software is available at http://sysbio.icm.edu.pl/memetic.

Keywords: Stochastic optimization, memetic algorithm, fuzzy clustering, statistical significance test.

1 Introduction

During the past three decades, Memetic Algorithms have been intensively studied in various areas [16]. Memetic Algorithms (MAs) are metaheuristics designed to find solutions from complex and difficult optimization problems. They are Evolutionary Algorithms (EAs) that include a stage of individual optimization or learning as part of their search strategy. In global optimization problems, evolutionary algorithms may tend to get stuck in local minima, and the convergence-rates of them are usually very low when there are numerous local optima. Thus, MA keeps local search stage to avoid it. The inclusion of a local search stage into the traditional evolutionary cycle of recombination-selection is a crucial deviation from canonical EAs.

Clustering [7] is a popular unsupervised pattern classification technique that partitions a set of n objects into K groups based on some similarity / dissimilarity metric where the value of K may ormay not be known *a priori*. Unlike

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 602–611, 2011.

hard clustering, a fuzzy clustering algorithm produces a $K \times n$ membership matrix $U(X) = [u_{k,j}]$, $k = 1,2,\ldots,K$ and $j = 1, 2,\ldots,n$ where $u_{k,j}$ denotes the membership degree of pattern x_j to cluster C_k. For probabilistic nondegenerate clustering $0 < u_{k,j} < 1$ and $\sum_{k=1}^{K} u_{k,j} = 1$, $1 \leq j \leq n$.

In our recent studies [11,10,12,15], we have seen that the fuzzy clustering techniques are depending on the choice of cluster validity measures and the exploration capability of the search space. These two facts motivated us to present a novel *Probabilistic Memetic Algorithm based Fuzzy Clustering* (PMAFC) technique that uses *a priori* probability of cluster validity measures to compute objective function. Moreover, the exploration capability has been increased by adaptive arithmetic recombination and opposite based local search techniques. The superiority of the proposed method over differential evolution based fuzzy clustering (DEFC) [11], genetic algorithm based fuzzy clustering (GAFC) [9], simulated annealing based fuzzy clustering (SAFC) [2] and FCM [6] has been demonstrated for four synthetic and two real life data sets. Also statistical significance test has been performed to establish the superiority of the proposed algorithm.

2 A Novel Probabilistic Memetic Algorithm Based Fuzzy Clustering

The proposed Probabilistic Memetic Algorithm based Fuzzy Clustering (PMAFC) technique has seven units to perform the fuzzy clustering. These units are described in below.

2.1 Representation of String And Population Initialization

Here the strings are made up of real numbers which represent the coordinates of the cluster centres. If string i encodes K cluster centres in d dimensional space then its length l will be $d \times K$. This process is repeated $\forall\ i = 1,2,\ldots,$NP of $S_i(t)$ strings in the population, where NP is the size of the population and t is the evolutionary clock (generation).

2.2 Computation of Objective Function

In PMAFC the objective function is associated with each string. The XB [17] and J_m [6] are two cluster validity measures chosen because they provide a set of alternate partitioning of the data. After computing XB and J_m for each string, the objective function, called total *a priori* probability, is computed as follow.

$$P_i(t) = \sum_{k=1}^{N} \sum_{i=1}^{NP} p(f_k(S_i(t))) \tag{1}$$

where, N is the number of functions or cluster validity measures and

$$p(f_k(S_i(t))) = \frac{value\ of\ f_k(S_i(t))}{Total\ value\ of\ f_j)} \ \forall\ k = 1, 2, \ldots, N\ and\ i = 1, 2, \ldots, NP \tag{2}$$

Hence, it is known that both XB and J_m measures are needed to be minimized in order to get good clustering. Thus, the objective function $P_i(t)$ has also been minimized.

Begin
1. t = 0; /* evolutionary clock (generation) is set to NULL*/
2. Rndomly generated an Initial Population $S_i(t)$ $\forall$ i = 1,2,. . .,NP;
3. Compute $f_k(S_i(t))$ $\forall$ k=1,2,. . .,N and i = 1,2,. . .,NP;
4. Compute *a priori* probability $p(f_k(S_i(t)))$ $\forall$ k=1,2,. . .,N and i = 1,2,. . .,NP;
5. Compute the total *a priori* probability $P_i(t) = \sum_{j=1}^{N} \sum_{i=1}^{NP} p(f_k(S_i(t)))$;
6. Set the *GBest(t)* by $S_i(t)$ where $P_i(t)$ is least;
For $t = 1$ **to** ***MAX-GENERATION*** **do**
For $i = 1$ **to** ***NP*** **do**
7. Compute μ_r = Rand(0,1);

$$8.\quad M_i(t) = \begin{cases} GBest(t) + S_i(t) & \text{if } \exp^{-(\frac{1}{t})} < \mu_r \\ S_i(t) & \text{otherwise} \end{cases}$$

9. Opposite based Local Serach for $M_i(t)$, store the result in $M_i^*(t)$;
10. Compute $f_k(M_i(t)$ and $f_k(M_i^*(t)$ $\forall$ k=1,2,. . .,N using *Step 3*;
End For
11. Combine all $f_k(S_i(t))$, $f_k(M_i(t))$ and $f_k(M_i^*(t))$, $\forall$ k=1,2,. . .,N and i = 1,2,. . .,NP, to get $f_k(S_j^*(t))$ $\forall$ k=1,2,. . .,N and j = 1,2,. . .,3NP;
12. Repeat *Step 4 and Step 5* to compute *a priori* probability and total *a priori* probability $P_j^*(t)$ $\forall$ j = 1,2,. . .,3NP;
13. Set the *GBest(t+1)* by $S_j^*(t)$ where $P_j^*(t) < P_j^*(t-1)$;
14. Generate $S_i(t+1)$, $\forall$ i = 1,2,. . .,NP, by selecting the best solutions are having lesser total *a priori* probability in $P_j^*(t)$ where j = 1,2,. . .,3NP;
15. t = t+1;
End For
16. Return best $s \in S_i(t-1)$ where i = 1,2,. . .,NP;
End

Fig. 1. Pseudocode Of PMAFC Algorithm

2.3 Recombination

Before the process of recombination, the best string of the current generation, called *GBest(t)*, is set by the solution that has least $P_i(t)$. Thereafter, for each $S_i(t)$, the adaptive arithmetic recombination is followed by Eqn. 3.

$$M_i(t) = \begin{cases} GBest(t) + S_i(t) & \text{if } \exp^{-(\frac{1}{t})} < \mu_r \\ S_i(t) & \text{otherwise} \end{cases} \tag{3}$$

The computation of offspring $M_i(t)$ is governed by $exp^{-(\frac{1}{t})}$. Note that as number of generation increases the value of $exp^{-(\frac{1}{t})}$ increases in the range between [0, 1] which results in a higher probability to create new offspring at the initial stage of generation by exploring the search space. Also the contribution of *GBest* to the offspring decreases with generation because that has already pushed the trail offspring to the global optimal. In Eqn. 3, μ_r is the recombination probability and that generated by $Rand(0, 1)$, a uniform random number generator with outcome $\in$ [0, 1].

2.4 Opposite Based Local Search

To explore the search space faster to create more offsprings, the opposite based local search is used. The concept of opposite point was introduced by Rahnamayana *et. al.* [13] and its defined as :

Definition. Let $M_i(m_1, m_2, \ldots, m_d)(t)$ be a point in the d-dimensional space, where $m_1, m_2, \ldots, m_d \in R$ and $m_k \in [x_k, y_k] \; \forall \in \{1, 2, \ldots, d\}$. The opposite point of $M_i(t)$ is defined by $M_i^*(m_1^*, m_2^*, \ldots, m_d^*)(t)$ where:

$$m_1^* = x_k + y_k - m_i \tag{4}$$

We used this definition for higher dimensional space to create more offspring $M_i^*(t) \; \forall \; i = 1,2,\ldots,\text{NP}$. Thus, the used technique is given the name as *opposite based local search*.

2.5 Other Processes

The generated current pool of offspring's ($M_i(t)$ and $M_i^*(t)$) are used to evaluate by $f_k(.) \; \forall \; k=1,2,\ldots,\text{N}$ and $i = 1,2,\ldots,\text{NP}$. Thereafter, all solution strings

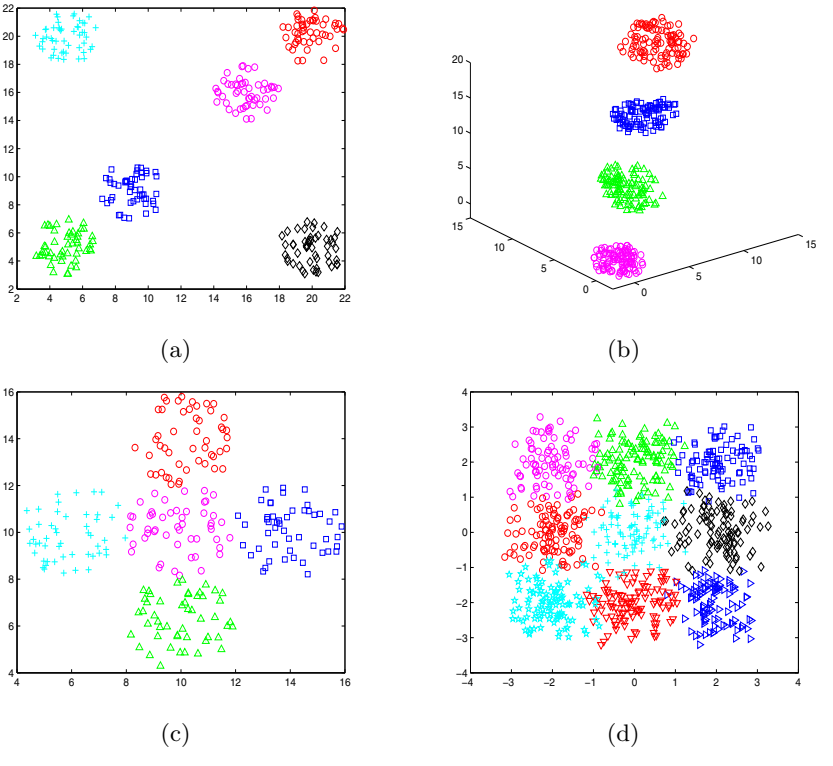


Fig. 2. True scattered plot of four synthetic data sets (a) AD_5_2_250, (b) AD_4_3_400, (c) Data_6_2_300, (d) Data_9_2_900

$S_i(t)$ $M_i(t)$, $M_i^*(t)$ and of its corresponding function values $f_k(S_i(t))$, $f_k(M_i(t))$, $f_k(M_i^*(t))$, $\forall$ k=1,2,. . .,N and i = 1,2,. . .,NP, are combined to create a new pool of solutions $S_j^*(t)$ and function values $f_k(S_j^*(t))$ where j = 1,2,. . .,3NP. The created $f_k(S_j^*(t))$ is now used to compute the total *a priori* probability $P_j^*(t)$ as described in Eqn. 1, $\forall$ k=1,2,. . .,N and i = 1,2,. . .,NP.

2.6 Selection

In the selection process, two operations have done. At first, the *GBest* is updated by comparing least $P_j^*(t)$ and $P_j^*(t-1)$. Thereafter, the $S_i(t+1)$, $\forall$ i = 1,2,. . .,NP, is generated by selecting the best solutions from $S_j^*(t)$, $\forall$ j = 1,2,. . .,3NP, which has least total *a priori* probability in $P_j^*(t)$.

2.7 Termination Criterion

All these units are executed for a fixed number of generation. The best string seen up to the last generation provides the solution to the clustering problem. The pseudocode of proposed algorithm is shown in Fig. 1.

3 Experimental Results

3.1 Synthetic Data Sets

AD_5_2_250: This dataset, used in [3], consists of 250 two dimensional data points distributed over five spherically shaped clusters. The clusters present in this data set are highly overlapping, each consisting of 50 data points. This data set is shown in Fig. 2(a).

AD_4_3_400: This dataset, used in [3], is a three dimensional data set consisting of 400 data points distributed over four squared clusters. This is shown in Fig. 2(b).

Data_6_2_300: This dataset, used in [4], contains 300 data points distributed over six clusters, as shown in Fig. 2(c).

Data_9_2_900: This dataset, used in [5], is a two dimensional data set consisting of 900 points. The data set has nine classes. The data set is shown in Fig. 2(d).

3.2 Real Life Data Sets

Iris: This data consists of 150 patterns divided into three classes of Iris flowers namely, Setosa, Virginia and Versicolor. The data is in four dimensional space (sepal length, sepal width, petal length and petal width).

Cancer: It has 683 patterns in nine features (clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses), and two classes malignant and benign. The two classes are known to be linearly inseparable.

The real life data sets mentioned above were obtained from the UCI Machine Learning Repository[1].

3.3 Input Parameters And Performance Metric

The proposed algorithm is adaptive in nature. Thus, we only need to provide the population size and number of generation and it is set to 20 and 100, respectively. Input parameters for DEAFC, GAFC and SAFC algorithms are same as used in [11,9,2]. The FCM algorithm is executed till it converges to the final solution. The performance of the clustering methods are evaluated by measuring Minkowski Score (MS) [8] and Silhouette Index (S(c)) [14].

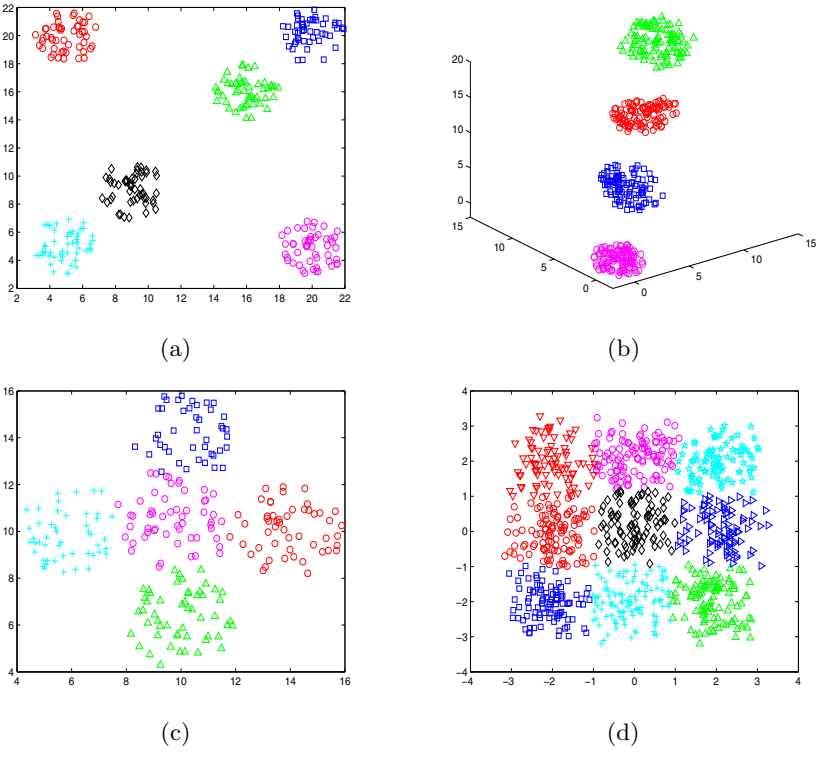


Fig. 3. Best scattered plot of four synthetic data sets (a) AD_5_2_250, (b) AD_4_3_400, (c) Data_6_2_300, (d) Data_9_2_900 after performing PMAFC

3.4 Results

The PMAFC algorithm is applied for four synthetic and two real life data sets to show the efficiency. Table 1 and Table 2 show the best performance of PMAFC algorithm in comparison of other well-known clustering techniques. However, each

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

Table 1. Best MS and S(c) values over 20 runs of different algorithms for four synthetic data sets

Algorithms	AD_5_2_250		AD_4_3_400		Data_6_2_300		Data_9_2_900	
	MS	S(c)	MS	S(c)	MS	S(c)	MS	S(c)
PMAFC	0.1803	0.7274	0.0000	0.8874	0.0000	0.9156	0.2004	0.7081
DEFC (XB)	0.2552	0.5817	0.1682	0.7531	0.1204	0.8417	0.4022	0.5604
DEFC (J_m)	0.2894	0.5463	0.1953	0.7202	0.1513	0.8104	0.3894	0.5937
GAFC (XB)	0.3147	0.5094	0.2171	0.6852	0.1864	0.7822	0.4591	0.5003
GAFC (J_m)	0.3329	0.4844	0.2447	0.6394	0.2104	0.7462	0.4432	0.5272
SAFC (XB)	0.3618	0.4483	0.2884	0.5875	0.2588	0.6864	0.4682	0.4407
SAFC (J_m)	0.3509	0.4372	0.3195	0.5404	0.2894	0.6505	0.4562	0.4822
FCM	0.3616	0.4174	0.3382	0.5173	0.3173	0.6227	0.5223	0.4053

Table 2. Best MS and S(c) values over 20 runs of different algorithms for two real life data sets

Algorithms	Iris		Cancer	
	MS	S(c)	MS	S(c)
PMAFC	0.1826	0.7108	0.0640	0.8284
DEFC (XB)	0.3546	0.4826	0.1804	0.6601
DEFC (J_m)	0.3794	0.4472	0.1659	0.6973
GAFC (XB)	0.3811	0.4226	0.2336	0.6001
GAFC (J_m)	0.4033	0.4092	0.2136	0.6482
SAFC (XB)	0.4219	0.3804	0.2781	0.5362
SAFC (J_m)	0.4407	0.3642	0.2592	0.5574
FCM	0.4603	0.3026	0.3214	0.5083

algorithm is evaluated by 20 consecutive runs. It is evident from the Table 1 and Table 2 that the PMAFC performs much better in term of Minkowski Score and Silhouette Index. For AD_4_3_400 and Data_6_2_300 data sets, PMAFC provides most optimal MS values (0.0), that means it performs the clustering perfectly. Similarly, for other data sets, it also shows better clustering results for both the performance measures. Fig 3 shows the best scatter plot for four synthetic data sets after performing PMAFC algorithm. By visual comparison of Fig 2 and Fig 3, it is quite clear that the PAFC algorithm provides similar structure for AD_4_3_400 and Data_6_2_300 data sets whereas others are almost similar with true structures. Please note that, colors and symbols used to represent the structure of true and predicted clusters are different because the labeling of points is different after applying the clustering method. However, their structures are similar. The boxplot representation of all algorithms is shown by Fig 4.

3.5 Test for Statistical Significance

It is evident from Table 1 and Table 2 that the MS values over 20 runs obtained by PMAFC algorithm is better compared to that obtained by other algorithms. Moreover, it has been found that for all the data sets, PMAFC provides the best MS values. In this article, we have used one way analysis of variance (ANOVA) [1] at the 5% significance level to compare the mean MS values produced by different algorithms in order to test the statistical significance of clustering solutions. Eight groups have been created for each dataset corresponding to the 8

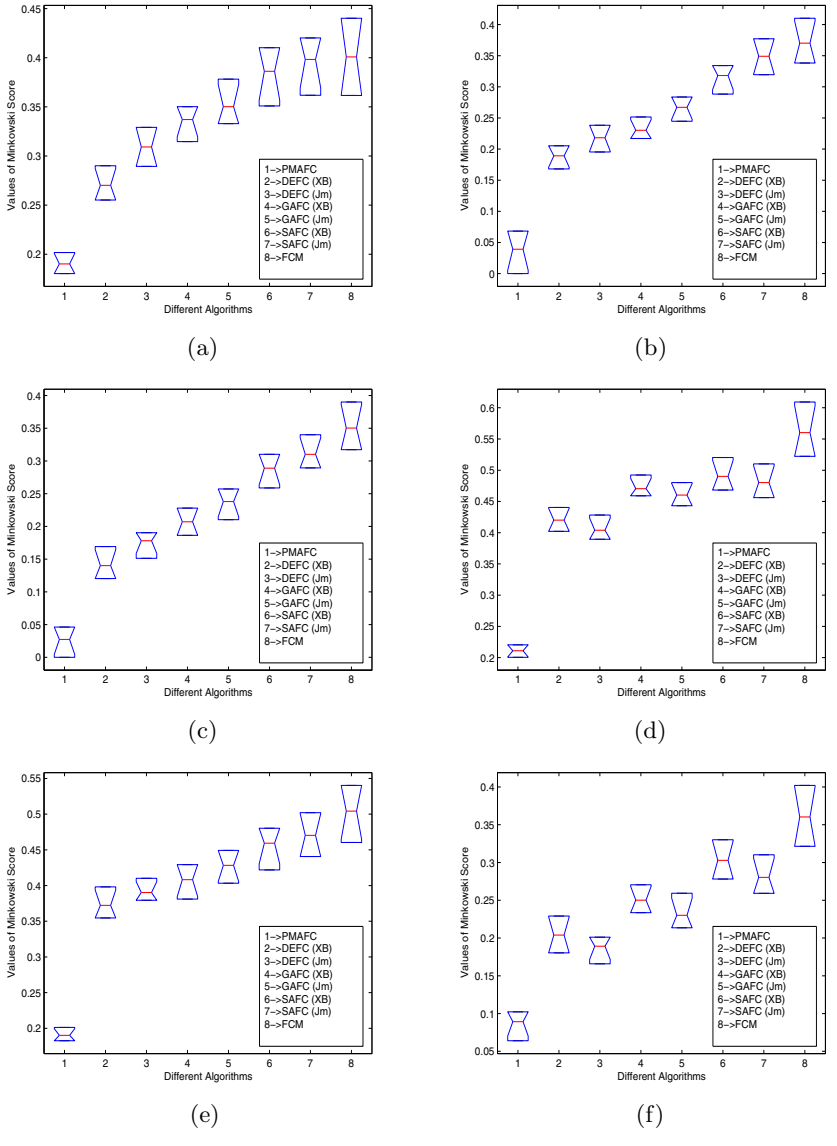


Fig. 4. Boxplot of MS for different clustering algorithm on (a) AD_5_2_250, (b) AD_4_3_400, (c) Data_6_2_300, (d) Data_9_2_900, (c) Iris (f) Cancer

algorithms viz., PMAFC, DEFC (XB), DEFC (J_m), GAFC (XB), GAFC (J_m), SAFC (XB), SAFC (J_m) and FCM. Each group consists of MS values obtained by 20 consecutive runs of the corresponding algorithm.

Table 3 shows the ANOVA test results for the six data sets used in this article. As the size of each group is 20 and there are total 8 groups, hence the degree of

Table 3. ANOVA test results for all the data sets comparing total 8 groups consisting of MS index scores of 20 consecutive runs of the 8 algorithms, i.e., PMAFC, DEFC (XB), DEFC (J_m), GAFC (XB), GAFC (J_m), SAFC (XB), SAFC (J_m) and FCM.

Data sets	df	F-critical	F-statistic	P-value
AD_5_2_250			103.8749	4.90668E-35
AD_4_3_400			222.9284	3.75138E-46
Data_6_2_300	158	2.1396	220.7108	5.28572E-46
Data_9_2_900			234.3181	6.77572E-47
Iris			172.1337	2.51864E-42
Cancer			141.2967	1.92075E-39

freedom (df) is 8 ×20 - 8 = 152. The critical value of F-statistic (The statistic used for ANOVA test) is 2.1396. It appears from Table 3 that the F-values are much greater than F-critical and the P-values are much smaller than 0.05 (5% significance level). These are extremely strong evidences against the null hypothesis which is therefore rejected for each dataset. This signifies that there are some groups whose means are significantly different.

4 Conclusion

In this article, a probabilistic memetic algorithm based fuzzy clustering technique has been described. The developed algorithm used *a priori* probability of cluster validity measures to compute the objective function. The cluster validity measure XB and J_m are used to get the clusters which are compact and well separable from each other. The exploration capability of the search space is increased by adaptive arithmetic recombination and opposite based local search techniques. The superiority of the proposed scheme has been demonstrated on a number of synthetic and real life data sets. Also statistical significance test has been conducted to judge the statistical significance of the clustering solutions produced by different algorithms. In this regard results have been shown quantitatively and visually.

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

References

1. Ferguson, G.A., Takane, Y.: Statistical Analysisin Psychology and Education, 6th edn. McGraw-Hill Ryerson Limited, New York (2005)
2. Bandyopadhyay, S.: Simulated annealing using a reversible jump markov chain monte carlo algorithm for fuzzy clustering. IEEE Transactions on Knowledge and data Engineering 17(4), 479–490 (2005)
3. Bandyopadhyay, S., Maulik, U.: Nonparametric genetic clustering: Comparison validity indices. IEEE Transactions on Systems, Man and Cybernetics, Part C 31(1), 120–125 (2001)

4. Bandyopadhyay, S., Maulik, U.: Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recognition 35, 1197–1208 (2002)
5. Bandyopadhyay, S., Murthy, C.A., Pal, S.K.: Pattern classification using genetic algorithms. Pattern Recognition Letters 16, 801–808 (1995)
6. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
7. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
8. Jardine, N., Sibson, R.: Mathematical Taxonomy. John Wiley and Sons, Chichester (1971)
9. Maulik, U., Bandyopadhyay, S.: Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. IEEE Transactions on Geoscience and Remote Sensing 41(5), 1075–1081 (2003)
10. Maulik, U., Bandyopadhyay, S., Saha, I.: Integrating clustering and supervised learning for categorical data analysis. IEEE Transactions on Systems, Man and Cybernetics Part-A 40(4), 664–675 (2010)
11. Maulik, U., Saha, I.: Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. Pattern Recognition 42(9), 2135–2149 (2009)
12. Maulik, U., Saha, I.: Automatic fuzzy clustering using modified differential evolution for image classification. IEEE Transactions on Geoscience and Remote Sensing 48(9), 3503–3510 (2010)
13. Rahnamayana, S., Tizhoosh, H.R., Salamaa, M.M.A.: A novel population initialization method for accelerating evolutionary algorithms. Computers & Mathematics with Applications 53(10), 1605–1614 (2007)
14. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Compt. App. Math. 20, 53–65 (1987)
15. Saha, I., Maulik, U., Plewczynskia, D.: A new multi-objective technique for differential fuzzy clustering. Applied Soft Computing 11(2), 2765–2776 (2010)
16. Tse, S.M., Liang, Y., Leung, K.S., Lee, K.H., Mok, T.S.: A memetic algorithm for multiple-drug cancer chemotherapy schedule optimization. IEEE Trans. Syst. Man Cybern. B 37(1), 84–91 (2007)
17. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 841–847 (1991)

An Approach to Intelligent Interactive Social Network Geo-Mapping

Anton Benčič, Márius Šajgalík, Michal Barla, and Mária Bieliková

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
{name.surname}@stuba.sk

Abstract. Map-based visualization of different kinds of information, which can be geo-coded to a particular location, becomes more and more popular, as it is very well accepted and understood by end-users. However, a simple map interface is not enough if we aim to provide information about objects coming from vast and dynamic information spaces such as the Web or social networks are. In this paper, we propose a novel method for intelligent visualization of generic objects within a map interface called IntelliView, based on an extensive evaluation and ranking of objects including both content and collaborative-based approaches. We describe the method implementation in a web-based application called Present, which is aimed at recycling items in social networks together with an experiment aimed at evaluation of proposed approach.

Keywords: social network, content filtering, map, personalization, visualization.

1 Introduction and Related Work

Intelligent presentation of data and information on the Web, including personalization is becoming crucial as the amount of available information increases in an incredible pace while the information is getting inter-connected in various different ways. It is becoming harder and harder to navigate in such a tangle of information resources of various kinds, origin and quality.

Special kind of information, which is gaining popularity nowadays, is information coming from social networks. Visualization of social networks is often realized in a form of complicated graphs that are hard to read and navigate in. Other approach is to use text interfaces to provide information about events happening within a social network. Neither of these approaches can keep pace with dynamics of social network and does not scale well as the size of the social network grows as well as the amount of information within it.

Most of the well-known approaches to social network visualization are not targeted at end-users, but rather on researchers or analysts, providing them with a quick

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 612–621, 2011.

overview of a social network and means for further statistical analysis and sociological research [7][1].

An example of a system, which is devoted to be used by end-users, is Viszter [2]. It visualizes a social network in a force-directed network layout. It focuses on using different visualization techniques to improve the navigation and search experience, but does not include any kind of content filtering or personalization, which would help to reduce the size of displayed network.

We decided to tackle the issue by realizing an intelligent map interface support for social networks visualization. A map, forming a basis of the visualization is a well known concept providing a basic level of information space partitioning based on geographical location. Location is often basic attribute of people or things involved in the social network. Another important partitioning is obtained by *intelligent evaluation of social network*, which partition the whole graph into several layers of abstraction and thus preventing the information overload problem.

There are many production-grade solutions based on map visualization, usually built on top of big online map services providers such as Google Maps or Microsoft Bing Maps. The system Geotracker [1] is interesting from our point of view, as it, apart from using the map interface to visualize geotagged RSS content, provides also a possibility to visualize development of the information space over time, which is similar to our idea of RealView described in this paper. Moreover, it contains basic user's interests model mapped to content categories, which is used for the personalization of the presentation layer.

In order to capture the dynamics of social networks, we also visualize important events and interactions. These are fully adapted to a user, to give him an overview of what happened since his last visit and to provide him with real-time updates as they happen.

Every visualization method obviously aims at presenting the most relevant content for a particular user. In the case when there are too many objects, which cannot be presented efficiently all at the same time, the first step that has to be exploited in one way or another is the content filtering. The first filtering that our approach called IntelliView performs is on a basis of the currently visible region of the map. However, this is far away from being enough, as the extent of a visible map can and by experience contains more content (objects) that can be presented at a time.

Because of this, our IntelliView makes another use of content filtering in a two-step process: (i) object evaluation and (ii) content presentation (visualization).

This paper is structured as follows. In first two sections we describe two basic steps of the proposed method for visualization: object evaluation (Section 3) and concept visualization (Section4). Section 5 is devoted to the evaluation of proposed method. Finally, we draw our conclusions.

2 Object Evaluation

The main issue of any kind of presentation-layer preprocessing is that if we want to get more accurate results, we need more data to collect and need to spend more time

[1] The best-known tools for visualizing and analyzing social networks are UCINET (http://www.analytictech.com/ucinet/) and PAJEK (http://pajek.imfm.si/).

per object for its evaluation. If we are using a lengthy evaluation method in a real world (i.e. the Web) environment with many objects to evaluate, this can be a problem as the users are willing to wait only a very limited time for the site or application to respond and if it does not, the user most often gives up.

An often used technique for overcoming the time problem is caching or pre-computing. Caching can help users speed up their consequent requests and is most useful in an environment where users follow a use pattern or tend to repeat their use-cases over a period of time. However, if there is large diversity in user behavior or the evaluation depends on time-varying parameters, the stored results become obsolete very quickly and the whole concept of caching fails. The technique of pre-computing has the same purpose and problems as caching with the difference that pre-computing is used for more complex computations that change even less often and their results are used for processing of each request.

Because our method for intelligent visualization of generic objects within a map interface needs to work with large sets of objects and present only a small portion of them at a time, we need more complex and thus time consuming computations to perform on every object. However, due to the aforementioned problem of computation and response times, we were forced to move some portions of the process into pre-computations stage.

Our object evaluation algorithm has both content-based and collaborative components. The final value for an object is assigned as follows:

$$k = c_o k_o + c_p k_p + c_c k_c$$

where c_o, c_p and c_c are constant coefficients that together sum up to one,

- k_o represents the index of general object significance,
- k_p is the index of personal object significance and
- k_c is the index of collaborative object significance.

The first two indexes are computed on request basis as they are content-based and may change more swiftly while the third, collaborative index is pre-computed on daily basis as its nature is much less dynamic.

2.1 General Object Significance

General object significance component tells how significant an object is to a random user without considering any personalization. It is computed as a sum of three subcomponents:

- Explicit feedback for the object coming from all system users.
- Object's lifetime within the system, with a negative correlation to object's significance (older objects are less significant than more recent ones).
- Category significance, assigned by an expert.

The three given parameters are put together in an equation:

$$k_o = c_{sup} d(\#of\ supports) + c_{cat} d(cat.sig.) - c_{tim} d(time)$$

The c_{sup}, c_{cat} and c_{tim} are coefficients that set the significance of single components and that sum up to one and determine the weight of aforementioned subcomponents

into the overall significance. The coefficients are set by a domain expert, but can be also discovered dynamically. In our representative implementation we started by manually estimating these coefficients according to the evaluation of a few representative test objects and subsequently these values were gradually fine-tuned dynamically and per-user by evaluating user's behavior and interaction with objects within the system as mentioned.

The equation also features a distribution function *d* that is an integral part of the content-based part of the evaluation algorithm. Its purpose is to distribute the outputs between zero and one in a way that the difference is most significant around the evaluated set's median. The distribution function is given by the following equation:

$$d(x) = \frac{tan^{-1}\left(\frac{x}{\frac{median(X)}{a}} - a\right) + tan^{-1}(a)}{\frac{\pi}{2} + tan^{-1}(a)}$$

The equation takes besides the input value two more parameters, which are constant throughout one iteration on a set of objects. The *median(X)* is a median value of the set of object parameters and the value *a* defines how sharp will be the edge between relaxed parts of the function and the steep part. The difference is shown in Figure 1, where both have the median of 100, the function in Figure 1a has *a=pi/2* and the function in Figure 1b has *a=4pi*.

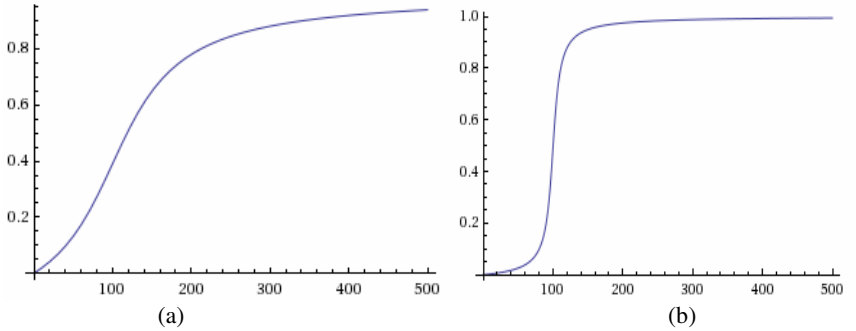


Fig. 1. Distribution function with (a) *a=pi/2,* (b) *a=4pi*

2.2 Personal Object Significance

The purpose of the personal object significance value is to encourage objects that the user might have interest in. The equation is as follows:

$$k_p = c_{cat} d(\text{category rank}) + c_{kword} d(top\ keyword\ rank)$$

The c_{cat}, c_{kword} and c_{sup} are again coefficients of component significances that sum up to one. These coefficients can be set in a similar manner as in general object significance calculation and in our representative implementation we set them in the

selfsame way. The category rank is a number derived from an ordered set of the user's N most popular categories.

It is expressed as:

$$\text{category rank} = \frac{N - [\text{category position}] + 1}{N}$$

The equation shows that the rank is determined by the category's position in the ordered set. The set always stores N most popular categories, where N is chosen by the environment's properties such as number of categories, number of users storage capacity and computation power, because these numbers need to be stored in a persistent storage and storing large number of categories for a large number of users would not be sustainable. The category's position is determined by the user's interest value that is raised every time user gives a positive implicit or explicit feedback to an object from that category while decreasing this value for all other categories. Once a category's value for a given user drops below a set threshold value, it is removed from the ordered set and a new category that the user developed an interest in recently can be stored. If a category is not contained within the set, its rank for the given user will be zero.

The top keyword rank works in a similar way to the category rank with the difference that it takes the top keyword contained within both object name, description, etc. and the table of top N keywords.

2.3 Collaborative Object Significance

Collaborative component of the evaluation function takes into account interests and activities not only of the user that the evaluation is meant for, but also users similar to him. Similar user is a fuser who has an interest in the same object categories and/or searches for the same keywords. As we are working with graph-like structures, we can take advantage of graph-based algorithms such as spreading activation [4], HITS [5] or PageRank [6] to find and rank similar and related users or objects.

When we find such a user, we assume that objects of his interest might be of interest to the current user as well. The evaluation is done in a similar way as the previous two components taking into account the level of similarity of the two users and the trustfulness of the similar user given by his rating.

3 Content Visualization

Due to enormous number of objects to be displayed, IntelliView has to choose only a small subset of them in order to maintain a lucid view. To accomplish this, objects are sorted by priority – a number that comes from the object evaluation described in the section 2. Prioritization is reflected by the most common visualization techniques that can be found in Bing Maps, Google maps or in a well-known ESRI's geographic information system (GIS) software products.

With the list of objects sorted by priority, IntelliView aims to achieve the following basic goals:

- displaying as much information as possible,
- displaying objects with higher priority first,
- illustrating the density of objects,
- maintaining a simple view.

We proposed two main methods for displaying objects, each of them with its own pros and cons. First, probably the most intuitive, method is depicted in Figure 2. There we can see a personalized view of Central Europe. Despite such a large area there in Slovakia emerges the local collection for people affected by the earthquake in Haiti. This is because of extensive donation of items made there by the user, his friends in social networks and other local users within that location. This first method has shown to be more successful and attractive to users than the second one.

In this method, objects are sorted and iterated over by their priority. For every object being iterated we check, whether there is or is not a collision with other objects, which are already displayed. If there is no collision, the object is displayed. If there is a collision, the object is considered to be displayed with less importance and in order not to overlay those already displayed (which have evidently higher priority), its size is reduced and opacity decreased. Ordered by importance, the objects are grouped into several levels of importance. Visually, displayed items differ based on the level, which they are included in.

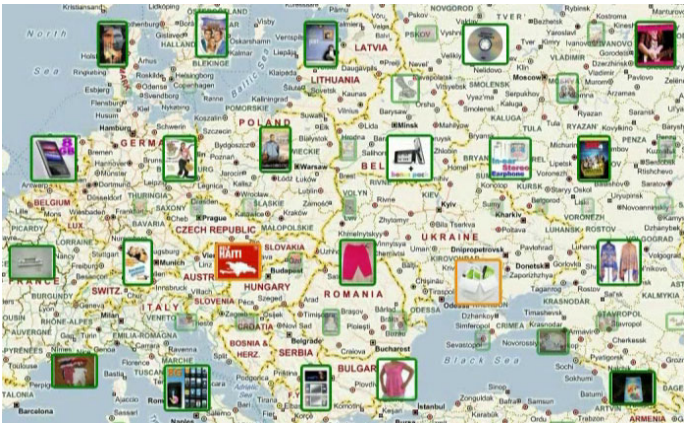


Fig. 2. Personalized visualization of IntelliView

To maintain lucid view, object distribution is pushed to the last level, which is not displayed. Thus, visual differences among levels can be less significant. Presence of hidden items in the last level is represented by background color of appropriate topmost item. In this manner, density can be viewed as well. Advantage of this method is in accurate projection of item location, but considering priority order, items in higher levels can have lower priority than those in lower levels.

The second method tries to achieve greater accuracy of displaying priority by dividing the whole viewport into a grid and considering items isolated, individually in each cell. Thus, each cell has its own representative, which is dominating in the center

of this cell. Items in lower levels of importance are displayed similarly as in preceding method around cell's representative. While having improved accuracy of priority order depiction, this method has also better performance in calculation following principles of divide and conquer.

However, in this case, location of item cannot be determined due to an error caused by the centralization within cell. The goal of centralization is to avoid collision problems on neighboring cell's bounds. As a positive side effect, due to uniform distribution of items on the map, this view is symmetric and more readable, but it may appear a bit artificial because of that. Finally, we abandoned this method due to its unattractiveness for users.

4 Real-time Visualization of Social Network Dynamics

If we want the users to understand the dynamics of the information space, which is being visualized, we need to clearly present the (relevant) transitions between various states within the information space. The RealView algorithm processes actions that other users perform within the viewing area and displays them as animations on the map.

The algorithm could just show all or at least majority of the actions, but it would not be practical for a couple of reasons. First of all with rising number of actions, the traffic increases as well. Second, the purpose of this dynamic view is to bring the user activity to encourage him to be active as well, but overwhelming him with animations would rather be counterproductive. That is why we implemented a filtering system, whose purpose is to select which actions will be delivered to the individual users.

The decision process is time sensitive and is based on a modified version of the relevancy system that is used in IntelliView. The time sensitivity means that despite of not showing all of the actions in the viewing area, the algorithm still has to be aware of the rate at which the actions happen. When implementing the mentioned features, we cannot work in true real-time. The basic idea is to stack the animations in a list, putting a delay interval between them that reflects the activity rate.

We briefly introduce two specific implementations of this concept. The first algorithm maintains an ordered list of actions and the number of actions from the last animation. Then every time a given number of actions are processed, it picks the first one and displays its animation.

The ordering is performed in a following way:

- when an action is intercepted, its relevancy value is calculated,
- all actions with lower relevancy are dismissed from the list,
- the processed action is added to the end of the list.

This ensures that the first action in the list has always the highest relevancy and that we keep all consequent animations.

The second algorithm is a simplification of the first one. It stores only one, most relevant action and the number of actions from the last animation. Then every time a given number of actions were processed it displays the stored animation.

5 Evaluation

In order to evaluate our approach to intelligent information visualization, we developed a web-based system called Present, which aims to help preventing precious nature resources by supporting re-use of ordinary goods and items instead of their endless production (see Figure 3). This is done by providing people with efficient means for donating or lending items, which are functional, but not used any more, and on the other side for requesting such donations or lendings.

Fig. 3. Present – introduction screen

Present is strongly connected to existing social networks such as Facebook, which play a crucial role not only in spreading the information about the system to new users, but also in providing a continuous motivation to use the system. This is achieved also by posting automatic updates to user's wall including photos of things he has provided, things he has borrowed or lent, events he has done, etc. and regular updates with user statistics, social impact information and posts about successful stories.

In addition to common statistics of the user activity, there are statistics provided by user geographical location. They include amount of welfare done in his location, statistics of users from his neighborhood and especially comparison with neighboring regions and countries. This indirectly leads to geo-competition and bigger interest in our system.

In order to evaluate the proposed approach, we filled the Present with artificial testing data about people, their items, donations and borrowings. The instances, albeit being generated artificially, were not completely random. We employed genetic

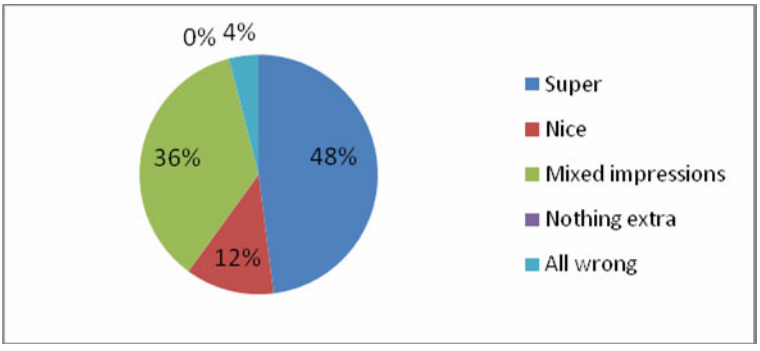


Fig. 4. Opinions of participants on the items and actions visualization on the screen and navigation among them

algorithm-like approach to perform crossovers between instances, so that a new instance would contain meaningful and credible attributes.

This way we were able to generate 2.7 million of virtual users placed on a map by following population density on our Earth. The system was running smoothly on an average personal computer in the role of a server even with such a great number of objects to be processed and displayed, which proves that the solution can be deployed in real-world conditions. Moreover, the real-world deployment could take advantage of cloud computing to distribute and handle the load.

Apart from basic evaluation of technical viability of the approach, we conducted a user-study with 25 volunteers, which were asked to use the system and provide us with feedback by filling-in a questionnaire. The majority of participants (76%) were attracted by the map interface immediately and 60% of them expressed that they were feeling comfortable when navigating on the map and found the layout of objects on the top of the map very well arranged (see Figure 4).

6 Conclusion

In this paper we presented a novel approach to visualization of large amounts of geo-coded and dynamic information, which can be nowadays found in social networks. Social data are getting more and more interest, as the Social Web spreads outside the environment of the isolated "social" applications and becomes a ubiquitous part of the ordinary Web. Traditional web-based information systems must take this into account and incorporate intelligent approaches in order to provide their users with the most relevant results while prevent information overload.

This is what we aimed at by our approach to information space visualization based on the intelligent map interface, which takes into account usage data of an individual as well as all users of system to determine user interests expressing what kind of content is relevant for the user and layout the content on the top of the map appropriately.

We evaluated the approach on the web-based system Present, which was filled with non-trivial amount of data and evaluated within a user study. The results showed

that the chosen visualization paradigm combined with the intelligent processing behind it provides an efficient means for personalized navigation within vast information spaces.

Acknowledgement. This work was partially supported by the grants VG1/0508/09, APVV-0208-10 and it is the partial result of the Research & Development Operational Programme for the project Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by the ERDF.

References

1. Gibbon, D., et al.: GeoTracker: Geospatial and Temporal RSS Navigation. In: Proc. of WWW 2007, pp. 41–50. ACM Press, Banff (2007)
2. Heer, J., Boyd, D.: Vizster: Visualizing Online Social Networks. In: IEEE Symposium on Information Visualization INFOVIS 2005, pp. 33–40. IEEE Press, Los Alamitos (2005)
3. MacEachren, A.M.: How maps work: representation, visualization, and design. The Guilford Press, New York (2004)
4. Crestani, F.: Application of spreading activation techniques in information retrieval. Artif. Intell. Rev., 453–482 (2004)
5. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications). Springer, New York (2006)
6. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. In: Proc. of the 7th International World Wide Web Conference, Brisbane, Australia, pp. 161–172 (1998)
7. Yang, X., Asur, S., Parthasarathy, S., Mehta, S.: A visual-analytic toolkit for dynamic interaction graphs. In: Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 2008, pp. 1016–1024. ACM, New York (2008)

Semantics of Calendar Adverbials for Information Retrieval

Delphine Battistelli[1], Marcel Cori[2], Jean-Luc Minel[2], and Charles Teissèdre[2,3]

[1] STIH / Paris Sorbonne University, 28 rue Serpente, 75006 Paris France
delphine.battistelli@paris-sorbonne.fr
[2] MoDyCo - UMR 7114 CNRS / Université Paris Ouest Nanterre la Défense, 200 av. de la République, 92001 Nanterre, France
{mcori,jminel,charles.teissedre}@u-paris10.fr
[3] Mondeca, 3 cité Nollez, 75018 Paris, France

Abstract. Unlike most approaches in the field of temporal expressions annotation, we consider that temporal adverbials could be relevant units from the point of view of Information Retrieval. We present here the main principles of our semantic modeling approach to temporal adverbial units. It comprises two steps: functional modeling (using a small number of basic operators) and referential modeling (using calendar intervals). In order to establish relationships between calendar zones, our approach takes into account not only the calendar values involved in adverbial units but also the semantics of prepositional phrases involved in these units. Through a first experiment, we show how Information Retrieval systems could benefit from indexing calendar expressions in texts, defining relevance scores that combine keywords and temporal ranking models.

Keywords: Information Retrieval, Temporal Information Semantic Modeling, Temporal Query.

1 Introduction

In this paper, we present our semantic modeling approach to temporal adverbial units, seen as relevant units both from the linguistic and the Information Retrieval point of view. Our application is based on the extraction and annotation of temporal adverbials found in texts, for which we provide first what we call a functional representation using a limited number of basic operators, followed by a referential representation using calendar intervals. We first describe (section 2) the context in which this research work took place. We then describe (section 3) our modeling approach, by illustrating how to transform a functional representation of a calendar adverbial into its referential counterpart. If the functional model is language dependant (in particular for the analysis of prepositions), the referential model is language independent. It means that this approach can be used in a multilingual system. Finally (section 4), we illustrate and discuss the benefits of the processing chain we are developing through an experiment which

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 622–631, 2011.

consists in the implementation of a simple search engine showing the type of results it is possible to obtain when combining thematic and calendar criteria in a query.

2 State of the Art

The importance of processing temporal information in texts with natural language processing techniques in order to upgrade applications in Information Retrieval (IR) is regularly emphazised. Among these applications, Question/ Answering systems, automatic summarization, search engines, and systems, either embedded or not, that display information on visual timelines (see for example [17]; [6], [3]) are mainly concerned.

Characterizing "*temporal information*" is in itself a challenge [13], both from a descriptive point of view (what are the language units which express temporal information?), and from an analytical point of view (what are the levels of representation and the computing strategies that need to be deployed in order to apprehend the semantic category of time?)

In the field of IR, temporal information is generally apprehended as what could assist in the resolution of a specific task: identifying the calendar anchors of the situations (generally named "events") described in texts, whether this task is seen as the ultimate goal or as a preliminary step needed to calculate event ordering. In the above-mentioned applications, linguistic expressions explicitly referring to a calendar system (for instance the Gregorian calendar), generally named "temporal expressions", have in particular been extensively researched. Moreover, in 2004, in the framework of Q/A systems, an evaluation task was organized, Time Expression Recognition and Normalization (TERN), exclusively targetted at the issue of the recognition and normalization (i.e. calculation of values with reference to a standard such as ISO) of this kind of expression. This task was the starting point for the approach aiming to standardize the semantic annotation of these expressions, the most popular propositions in this field being TIMEX2 [10] and TIMEX3 [16]. This led to the elaboration of annotated corpora such as ACE and TimeBank[1] and several automatic systems dealing in particular with the computing process of relative (deictic or anaphoric) temporal expressions (see for example [11]; [1]; [15]).

Regardless of the difficulties such systems had to deal with, none of these approaches aimed at recognizing and annotating whole adverbial units as temporal expressions. In fact, it is not the adverbial expressions which are annotated (see TIMEX2 [10] and TIMEX3 [16]), but only their reference to a calendar system. In the expression "in 1992", for example, the preposition "in" is annotated outside the tag <TIMEX3> by the tag <SIGNAL>. Whatever the underlying linguistic theory, it follows that approaches based on this kind of scheme do not take advantage of the different sorts of relations between temporal expressions which can be revealed by the semantic analysis of prepositional phrases.

[1] See corpora LDC2005T07 and LDC2006T06 for ACE; and LDC2006T08 for TimeBank in the LDC catalogue (http://www.ldc.upenn.edu).

Our approach, in contrast (see section 3), aims to analyse and to formalize the way in which the reference to time is expressed in the language by adverbial units. Furthermore, this approach reveals a pertinent search mode for information that depends on temporal criteria (see section 4). It is possible to express a query such as "What happened at the beginning of the 80's?", since the system is able to compute a relation between the expression "at the beginning of the 80's" and expressions found in texts such as "on January 10 1985" or "from January 10 1985", leaving it up to the user (or to a dedicated system) to identify the events connected with these temporal zones. This kind of approach belongs to the same research trend as that which attempts to take advantage of annotated corpora, such as the 2010 HCIR edition which tackles the issue of skimming through the New York Times [4,14].

3 From Calendar Expressions to Calendar Intervals

The corner stone of this research project consists in describing the semantics of temporal adverbials as decomposable units calling upon a compositional or functional analysis and in providing a methodology to transform this functional representation into its calendar counterpart. We consider *Calendar Expressions*, which may have one of the following forms:

on January 10 1985
at the beginning of the 80's
three months before the beginning of the 80's
until three months before the beginning of the 80's

3.1 Functional Representation

Except for the approach presented in [5], only a few research projects have shown interest in describing the semantics of temporal adverbials as decomposable units calling upon a compositional interpretation of their significance. This is the approach adopted here. Furthermore, a linguistic study has led us to consider calendar adverbial units as a conjunction of operators interacting with temporal references [7]. Thus our approach insists on how language refers to and designates areas on a calendar[2].

The functional representation of calendar expressions analyses them as a succession of operators operating on a *Calendar Base (CB)*. The Calendar Base is the calendar reference indicated by a calendar expression (calendar temporal units, such as "the 17th century", "June 6", or time periods such as "decade", "month", "hour", for instance). Several operators are successively applied upon the Calendar Base: (i) Pointing Operators ("last year", "this month"), (ii) Zooming and Shifting Operators ("mid 80s", "three months before"), and (iii) Zoning Operators ("before", "until", "around").

[2] See [12] for a reasoned perspective about the pertinence of dealing with temporal adverbials as basic units when analysing time in language.

Each of those operators can express various semantic values that the functional representation transcribes in a controlled tagset. These values provide information on how temporal references can be interpreted and which transformation procedures should be used to handle them in order to end up with proper calendar intervals. The functional representation reflects the structure of calendar expressions which (for unary instances) can be represented through the following generic form:

```
OpZoning(OpZooming/Shifting(OpPointing(CB))
```

(1) Calendar bases are composed of cardinal and ordinal temporal units (minute, hour, dayOfWeek, dayOfMonth, month, trimester, semester, year, etc.).
(2) Pointing operation: this operation specifies whether an expression is absolute or relative. In accordance with traditional approaches, "absolute"expressions which can be directly located in the calendar system ("in 2010", "during the 70s") are distinguished from "relative"expressions which require computation to be anchored in the calendar system ("two years before", "yesterday"). The values of pointing operations also specify how to interpret "relative"calendar expressions, distinguishing those which refer to another temporal reference given by the text (anaphoric expressions such as "two days later", "the next day") and those which refer to the time of the utterance situation (such as "tomorrow"or "the past few months"). Absolute expressions are those which are built on a *complete* calendar base, i.e. a well defined one.
(3) Shifting and zooming operation: the zooming operation encodes qualitative focus shifts, for expressions such as "by the end of the month"or "in the early nineties". More precisely, the zooming operations are, ZoomID (identity) and:

(3.1) ZoomBegin (Zb)
(3.2) ZoomEnd (Ze)
(3.3) ZoomMiddle (Zm)

The expression "at the beginning of the 80s" would lead to the following analysis[3]:

```
ZoningID(ZoomBegin(CB(decade: 1980)))
```

The possible values of the shifting operation describe a temporal shift that operates on a calendar base, for expressions such as "two months before our century", or "next month". The information specified is: (i) the shifting orientation (is it backward or forward?); (ii) and, when necessary, the temporal granularity of the shifting operation (day, month, etc.) and its quantity.
Shifting operations are:

(3.4) ShiftBefore Sb$(v, -n)$
(3.5) ShiftAfter Sa$(v, +n)$

For instance, the functional representation of the expression "two months after the beginning of the year 1985" would be:

[3] Where ZoningID is the identity zoning operator, see below.

```
ZoningID((Shift(month,+2))(ZoomBegin(CB(year: 1985))))
```

For the expression "three months before the beginning of the 80's", it would be:

```
ZoningID((Shift(month,-3))(ZoomBegin(CB(decade:1980))))
```

(4) Zoning operation: this operation specifies the semantics of units such as since, until, before, after, around or during. Briefly, the functional analysis starts from a calendar base definition that acts as a core reference, and progresses toward the final zoning operation which specifies the calendar area that is finally designated by the whole calendar expression. In the above examples we have already used the identity operation (ZoningID). The other operations are:

(4.1) Be (before).
(4.2) Af (after).
(4.3) Un (until).
(4.4) Si (since).
(4.5) Binary operator Between

Thus, for the expression "until three months before the beginning of the 80's", we get:

```
ZoningUntil ((Shift(month,-3))(ZoomBegin(CB(decade: 1980))))
```

And for "between the end of the year 2007 and the beginning of March 2009":

```
BETWEEN( ZoningID(ZoomEnd(CB(year: 2007))),
         ZoningID(ZoomBegin(CB(month: 3, year: 2009))) )
```

3.2 Referential Representation

Here we propose a referential representation of calendar expressions, in terms of *Calendar Intervals (CI)*.

Calendar Units. We take a finite set of *units* $U = \{u, v, w, ...\}$. For example: {*millennium, century, decade, year, month, day, ...*}. To each unit u is associated an infinite sequence:

$$S(u) = \langle \ldots, u_{-n}, \ldots, u_{-1}, u_0, u_1, \ldots, u_m, \ldots \rangle$$

$S(u)$ describes the succession of dates according to a given unit. For example, if u is the *month*, $S(u)$ will be a sequence such as:
$\langle \ldots$, 2010-11, 2010-12, 2011-1, 2011-2, $\ldots \rangle$

We also define an order relation[4] between units: we say that unit u is smaller than unit v, and we write $u \leq v$. In this case we define a mapping $f_{v \to u}$ such that an ordered pair $\langle v_j, v_k \rangle$ is associated with each u_i, in other words a pair $\langle j, k \rangle$ is associated with each i, such that:

[4] In this paper we consider that this relation is a total order. In further work we will treat the case of a partial order and will consider *seasons, weeks, nights,....*

(i) $j \leq k$
(ii) if $i_1 < i_2$, if $f_{v \to u}(i_1) = \langle j_1, k_1 \rangle$, if $f_{v \to u}(i_2) = \langle j_2, k_2 \rangle$, then $k_1 < j_2$
In particular, for each u, $f_{u \to u}$ is such that:

$$\forall i \; f_{u \to u}(i) = \langle i, i \rangle$$

This means that a corresponding interval in a smaller unit is associated to a punctual date, and that temporal order is respected in the passage from one to the other. For example, if v is the year and u is the *day*, we have:
$f_{v \to u}(2010) = \langle$2010-1-1, 2010-12-31$\rangle$

Calendar Intervals. A *Calendar Interval* (or CI) is given by an ordered pair of elements taken from one of the sequences $S(u)$: $\langle u_i, u_j \rangle$ (with $i \leq j$). We can also write: $\langle i, j, u \rangle$. Particular cases where $i = -\infty$ ou $j = +\infty$ will be admitted.

u_i represents the date of the beginning, u_j represents the date of the end and u is the unit. For each CI $\langle u_i, u_j \rangle$ where the unit is u and for each v smaller than u we can associate an *equivalent* CI: if $f_{u \to v}(i) = \langle i_1, j_1 \rangle$, if $f_{u \to v}(j) = \langle i_2, j_2 \rangle$, then the equivalent CI is $\langle v_{i_1}, v_{j_2} \rangle$. For example, to the CI $\langle$1995-3, 1996-5$\rangle$ we can associate the equivalent CI $\langle$1995-3-1, 1996-5-31$\rangle$.

CI Associated to a Functional Expression. Given a functional expression, we define a translation process, step by step, in a Calendar Interval.

(1) To each complete calendar base we associate a CI or, to be more precise, an element u_i, then, by application of $f_{u \to u}$, a CI $\langle u_i, u_i \rangle$. For example:
January 1985: $\langle$1985-1, 1985-1$\rangle$
January, 10 1985: $\langle$1985-1-10, 1985-1-10$\rangle$
The 80s: $\langle$198_, 198_$\rangle$

A functional expression is obtained by the successive application of operators to a complete calendar base. Simultaneously, we apply operations to the CIs, starting with the CI associated to the calendar base. This makes it possible to associate a computed CI to each calendar expression.

(2) A pointing operator can give a complete calendar base if none exists at the origin.

(3) Except for the identity, which produces no transformation, the zooming/ shifting operators give the following transformations:

(3.1) ZoomBegin (Zb): we define a coefficient τ (taken between 0 and 1). This coefficient may depend on the expression: *at the beginning, at dawn, in the early beginning,....* It can also depend on the desired degree of precision. In the following we will take $\tau = 0.25$. To $\langle u_i, u_j \rangle$ we associated, for each unit v strictly smaller than u, the CI:

$$\langle f_{u\to v}(i), f_{u\to v}(i) + \lceil \tau(f_{u\to v}(j) - f_{u\to v}(i) + 1)\rceil, v\rangle$$

Thanks to the ceiling function[5], we always obtain integers. Consequently, the result will be different depending on the unit taken into account. So, for *at the beginning of the 80's*, we obtain:

⟨1980, 1982⟩
⟨1980-1, 1982-6⟩
⟨1980-1-1, 1982-7-2⟩

(3.2) ZoomEnd (Ze): we use a coefficient τ' (which can be equal to τ). To $\langle u_i, u_j\rangle$ we associate, for each unit v strictly smaller than u, the CI:

$$\langle f_{u\to v}(j) - \lceil \tau'(f_{u\to v}(j) - f_{u\to v}(i) + 1)\rceil, f_{u\to v}(j), v\rangle$$

(3.3) ZoomMiddle (Zm): we define a new coefficient μ (taken between 0 and $\frac{1}{2}$). To $\langle u_i, u_j\rangle$ we associate, for each unit v strictly smaller than u, the CI:
$\langle f_{u\to v}(i) + \lceil \mu(f_{u\to v}(j) - f_{u\to v}(i) + 1)\rceil, f_{u\to v}(j) - \lceil \mu(f_{u\to v}(j) - f_{u\to v}(i) + 1)\rceil, v\rangle$

(3.4) ShiftBefore Sb$(v, -n)$, where v is a unit of U smaller than u (or equal to u). If $f_{u\to v}(i) = \langle k, l\rangle$, to $\langle i, j, u\rangle$ we associate the CI $\langle k - n, k - n, v\rangle$. So, for *three months before the beginning of the 80's*, we obtain: ⟨1980-10, 1980-10⟩.

(3.5) ShiftAfter SA$(v, +n)$,where v is a unit of U smaller than u (or equal). If $f_{u\to v}(j) = \langle k, l\rangle$, to $\langle i, j, u\rangle$ we associate the CI $\langle l + n, l + n, v\rangle$.

(4) Zoning operators produce transformations:
(4.1) Be (before). To $\langle i, j, u\rangle$, we associate $\langle -\infty, i - 1, u\rangle$.
(4.2) Af (after). To $\langle i, j, u\rangle$, we associate $\langle j + 1, +\infty, u\rangle$.
(4.3) Un (until). To $\langle i, j, u\rangle$, we associate $\langle -\infty, j, u\rangle$.
So, for *until three months before the beginning of the 80s*, we obtain:
⟨−∞, 1980-10⟩
(4.4) Si (since). To $\langle i, j, u\rangle$, we associate $\langle i, +\infty, u\rangle$.
(4.5) Binary operator Between. It applies to two CIs $\langle i_1, j_1, u\rangle$ and $\langle i_2, j_2, v\rangle$. We consider w, the largest unit smaller than u and v. We assume that $f_{u\to w}(j_1) = \langle k_1, l_1\rangle$, and that $f_{v\to w}(i_2) = \langle k_2, l_2\rangle$. Thus we obtain: $\langle l_1 + 1, k_2 - 1, w\rangle$. For example, in order to represent *between the end of year 2007 and the beginning of March 2009*, we obtain: ⟨2008-1-1, 2009-2-28⟩.

Irrelevant Cases. There are some cases that the model does not attempt to deal with. For example:
(4.6) Outside. This operator involves considering the union of CIs: to $\langle i, j, u\rangle$, would be associated $\langle -\infty, i - 1, u\rangle \cup \langle j + 1, +\infty, u\rangle$.
Similarly, cases such as *every Monday in January 2010* would be represented by 4 CIs.

[5] $\lceil x\rceil$ designates *ceiling*(x).

3.3 Toward a Heuristic to Calculate Calendar Expressions Relevance

Based on the referential representation, the aim is to provide criteria to compute similarity scores between calendar expressions and an end-user's query containing calendar data. The ultimate goal of this Information Retrieval service is to facilitate the exploration of text corpora content. While relationships between time intervals as described by [2] allow comparisons of calendar expressions (by testing inclusion, overlapping, etc.), they do not provide information on how to rank calendar expressions by relevance, which is the core of our approach.

A weighting function, which combines the intersection between a potential answer and the query and their temporal distance, has been used to compute a final temporal relevance score. It is described in details in [8].

4 Experimentation

In a specific use case we are working on, the resources are part of a process that adds metadata to texts in order to provide structured data for an Information Retrieval system, which is parameterized to facilitate access to calendar information. The experiment consists in a simple search engine capable of handling complex calendar queries. Indexation and querying are based on Apache Lucene software[6]. The processing chain is composed of several modules: (1) calendar expression annotator, (2) calendar model transformation module, (3) interval relations indexer and search module. The calendar expressions annotator module [18] provides an annotated text that conforms to the functional model described in section 3.1 (recall rate: 84.4%, precision rate: 95.2%[7]). At present, these resources annotate texts in French. The calendar model transformation module is a direct implementation of the transformation algorithm described in section 3.2.

Once calendar expressions have been transformed into their calendar interval counterparts, it becomes possible to describe their relationships in term of relevance. Indexing methodology enables thematic search criteria (keywords) and temporal criteria to be combined, by aggregating relevance scores. Sentences rather than entire documents are indexed, in order to display text segments containing temporal expressions that answer a query, but also to ensure that the distance between the calendar expression and the keywords searched, if found in the text, is not too great. The requests submitted by users to the system are annotated with the same annotation module that is used to annotate calendar expressions in the indexed documents. The analysis of the query separates a set of keywords (the thematic query) and the temporal query, expressed in natural language.

For this first experiment, around 8.000 French articles from Wikipedia relating to the history of France were annotated and indexed. Figure 1 is a screenshot of

[6] `http://lucene.apache.org/`

[7] At this stage of our development, only "absolute" expressions (which do not require specific computation to be anchored in the calendar system) are taken into account.

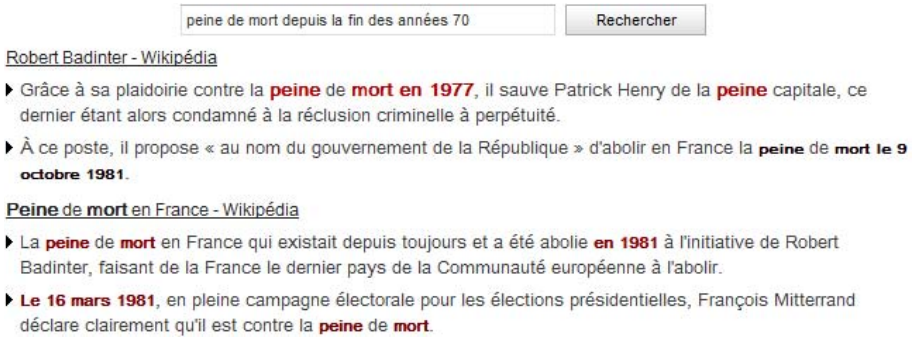


Fig. 1. Experimental temporal-aware search engine screenshot

the results returned for the following query *"peine de mort depuis la fin des annes 70"* (*"death penalty since the end of the 70s"*). The top results are those whose calendar expression is considered semantically similar to the query. We see that the semantics of the calendar expressions both in the query and the indexed documents has been properly interpreted, since the resulting expressions ("in 1977", "on October 9, 1981", "in 1981", "on March 16, 1981") match with the period specified in the query. It is also important to note that the temporal distance between the documents and the query affects the results ranking. The results are ordered from most to least relevant (for each documents) considering the period defined by the query. We are currently working on the specifications of a protocol in order to evaluate the query system.

5 Conclusion

The processing chain described here aims to show how our principles could be useful for systems that require temporal querying. Our application evaluates the semantic distance/similarity between temporal expressions, in order to provide the most relevant expressions to answer a temporal query. This work is complementary with other research projects, such as automatic feeding of timelines and text navigation software [9]. The algorithm for temporal expressions ranking is not only designed for search engines but can also be used in other tools that involve Temporal Information Retrieval. With this aim in view, we are currently working on a text navigation system designed to facilitate access to temporal information in texts.

References

1. Ahn, D., van Rantwijk, J., de Rijke, M.: A cascaded machine learning approach to interpreting temporal expressions. In: Proc. NAACL-HLT 2007, Rochester, NY, USA (April 2007)
2. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26, 832–843 (1983)

3. Alonso, O., Gertz, M., Baeza-Yates, R.: Clustering and Exploring Search Results using Timeline Constructions. In: Proc. CIKM 2009, Hong Kong (2009)
4. Alonso, O., Berberich, K., Bedathur, S., Weikum, G.: Time-based Exploration of News Archivies. In: 4th HCIR Workshop, New Brunswick, NJ, August 22 (2010)
5. Aunargue, M., Bras, M., Vieu, L., Asher, N.: The syntax and semantics of locating adverbials. Cahiers de Grammaire 26, 11–35 (2001)
6. Barzilay, R., Elhadad, N., McKeown, K.R.: Inferring stategies for sentence ordering in multidocument news summarization. Journal of Artificial Intelligence Research 17, 35–55 (2002)
7. Battistelli, D., Couto, J., Minel, J.-L., Schwer, S.: Representing and Visualizing calendar expressions in texts. In: Proc. STEP 2008, Venice (September 2008)
8. Battistelli, D., Cori, M., Minel, J.-L., Teissèdre, C.: Querying Calendar References in Texts, 8 (submitted)
9. Couto, J., Minel, J.-L.: NaviTexte, a text navigation tool. In: Basili, R., Pazienza, M.T. (eds.) AI*IA 2007. LNCS (LNAI), vol. 4733, pp. 720–729. Springer, Heidelberg (2007)
10. Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G.: TIDES 2005 Standard for the Annotation of Temporal Expressions (2005), `http://www.ldc.upenn.edu/Projects/ACE/docs/English-TIMEX2-Guidelines_v0.1.pdf`
11. Han, B., Gates, D., Levin, L.: From language to time: A temporal expression anchorer. In: Proc. TIME 2006, pp. 196–203 (June 2006)
12. Klein, W.: Time in Language. Routledge, London (1994)
13. Mani, I., Pustejovsky, J., Gaizauskas, R. (eds.): The Language of Time. A Reader. Oxford Linguistics, Oxford University Press Inc., New York (2005)
14. Matthews, M., Tolchinsky, P., Blanco, R., Atserias, J., Mika, P., Zaragoza, H.: Searching Through Time in the New York Times. In: Proc. HCIR 2010, New Brunswick, NJ, August 22 (2010)
15. Mazur, P., Dale, R.: WikiWars: A New Corpus for Research on Temporal Expressions. In: Proc. 2010 Conference on Empirical Methods on Natural Language Processing, October 9-11. MIT, Massachusetts (2010)
16. Pustejovsky, J., Castano, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust Specification of Event and Temporal Expressions in Text. In: Proc. IWCS-5 Fifth International Workshop on Computational Semantics (2003)
17. Saquete, E., Martínez-Barco, P., Muñoz, R., Vicedo, J.L.: Splitting Complex Temporal Questions for Question Answering systems. In: Proc. ACL 2004, Barcelona, Spain (July 2004)
18. Teissèdre, C., Battistelli, D., Minel, J.-L.: Resources for Calendar Expressions Semantic Tagging and temporal Navigation through Texts. In: Proc. LREC 2010, Malta, May 19-21 (2010)

Concentric Time: Enabling Context + Focus Visual Analysis of Architectural Changes

Jean-Yves Blaise and Iwona Dudek

FRE 3315 CNRS/MCC,
13288 Marseille Cedex 09, France
{jyb,idu}@map.archi.fr

Abstract. In order to acquire and share a better understanding of architectural changes, researchers face the challenge of modelling and representing events (cause and consequences) occurring in space and time, and for which assessments of doubts are vital. This contribution introduces a visualisation designed to facilitate reasoning tasks, in which a focus view on evidence about what happens to artefact λ at time t is a complemented with a context view where successive spatial configurations of neighbouring artefacts, durations of changes, and punctual events are correlated and tagged with uncertainty markers.

1 Introduction

Analysing, representing, cross-examining what is *really* known about transformations occurring during the lifetime of historic artefacts is a research topic where issues specific to historic sciences often meet models and formalisms developed in computer science. Although largely outnumbered by self-evident applications of computer graphics, fruitful knowledge representation-oriented research efforts do exist on the modelling of such spatial dynamics.

But *gaining insight* (in the sense of [1]) into architectural changes is not only about circumscribing the relevant pieces of knowledge – *i.e.* abstracting relevant properties of the reality observed, and foreseeing possible processing. In historic sciences particularly, it is also basically about giving users means to shed a new light on their knowledge, to draw individual, self-achieved, maybe temporary conclusions – *i.e. graphics as a discovery tool* [2]. And indeed an effective way to support reasoning is to capitalize on the user's capacity to *use vision to think* [3].

Now what do we here need to reason about? Architectural changes occur in space, and in time – analysing changes will mean handling both dimensions. Changes may have implications on the shape of an artefact or not, anyway the evidence is liable to be distributed in space/time slots. Of course, visual solutions that help distributing and analysing information in space exist, either in 2D or 3D [4]. In parallel, other visual solutions may help us to distribute events in time, using the basic timeline paradigm, or more sophisticated concepts [5][6]. But architectural changes do not occur in space, on one hand, and in time, on another hand. If we want graphics to help us perform reasoning tasks, we need them to combine a spatial representation and a

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 632–641, 2011.

chronology representation[1]. So-called "time sliders" (nested inside 2D or 3D interactive graphics) could be seen as an answer: they allow users to select a time slot and observe the corresponding spatial configuration. But what the user sees is one period at a time: in other words, a focus view. And beyond merely pointing out facts – artefact λ changes at time t – *amplifying cognition* [7] about architectural changes implies a teleological approach through which we connect facts about an artefact to a context – previous and following changes, things that happen in the neighbourhood, events that occur during that same period of time, *etc.* Besides a focus view, we need a context view where successive spatial configurations, durations of changes, punctual events, *etc.*, are correlated. Finally, as usual in historic sciences, the evidence we base on is uncertain: efficient graphics should help visualising that uncertainty[2].

In previous contributions, we focused on modelling issues, starting from two ends: the spatial bias (modelling the lifetime of artefacts as chains linking version to one another, with identification/classification issues) and the temporal bias (modelling key moments in the evolution of artefacts and processes of transformation in a cause + consequence approach). As a result, we developed solutions ranging from knowledge visualisation to information systems where either spatial interfaces or timeline-like graphics helped visualise architectural changes [8][9]. But at the end of the day, although our KR effort did integrate time and space, means to perform visual analysis privileged *either* time *or* space. In this research we present an experimental visualisation– called concentric time – designed according to four simple ideas:

- a combination of spatial features and chronology allowing comparisons enforced *within the eyespan* [10],
- an application of the context+focus principle,
- a support for uncertainty assessment (fuzziness, impreciseness, lacks, etc),
- a *simple* (in the sense of [11]) visualisation minimising the learning curve and the decoding effort, facilitating information discovery (including by a public of non-experts).

It is applied and evaluated on the market square in Krakow, Poland, a 200*200m urban space where 24 artefacts have been built, modified, and for most of them destroyed, over a period of 750 years. Primarily designed as a visualisation, it also acts as an interface. Section 2 details where we start from. Section 3 presents the concentric time visualisation itself. In section 4, its evaluation is discussed. Finally, in sections 5 and 6, we point out some of its limits, and further possible investigations.

---

[1] Classic gothic cathedrals in Bourges or Chartres were built in some decades, starting from the end of the XIIth century. The construction of the gothic cathedral in Tours started at the same period, but lasted until the XVIth century. And it is precisely that difference in time that explains some of the differences in the shapes of these edifices, *i.e.* differences in space.

[2] The information can be imprecise to the extent that although an edifice λ did exist between time $t1$ and $t2$ in a given "area", we cannot precisely position it inside this area. However, unlocalised **e** may have had an influence on others around. For instance, in our case study, an edifice called "Smatruz" (an open market hosting traders temporarily), localised "*probably somewhere in the south west corner*", was destroyed in mid XVth century. Its function being still needed, it is apparently drifted to nearby "Sukiennice", causing changes on this latter edifice, *i.e.* no reasoning on "Sukiennice" without representing un-localised "Smatruz".

2 Research Background

Combining spatial information and a chronology inside one unique representation is far from being a new problem. In a way, it is a feature of Roman "ribbon maps", where localities across the empire are connected through lines that mean travel durations, and not actual distances. Far before the "computer era", XIXth century scientists developed inventive and sometimes brilliant solutions like Minard's "figurative maps", to this day still regarded as exemplary [12]. John Snow's analysis of the 1854 London cholera epidemic, as masterly analysed by E.R Tufte [10], shows how reasoning both on time and space, in a casual manner, can been vital. Yet we today expect from graphics features that paper-based solutions cannot offer (updatability, interactive browsing, interface capabilities, etc.). Let us still remember to counterbalance the natural verbosity and jumble-hungriness of computer-based solutions with the clarity of mind of the above mentioned precursors.

In this section we briefly present some of the prominent solutions to combine space and time explicitly, at visualisation level. Our intent is not exhaustiveness – unreachable here (see for instance [4]) – but to say basically where we start from.

Originating from geosciences for the former, from engineering sciences for the latter, GIS platforms and CAD tools can allow users to display at a given position ***p*** successive versions of an object using the "layer" concept (although this concept may take a different name depending on the actual software). And indeed, the layering of time-related info may be efficient in terms of information delivery. Each version of the object can correspond to a given time slot, and so the job is done – space and time are present. Well apparently the job is done, but apparently only: when giving a closer look time does not really exist here as such (with for instance vital parameters such as durations of changes not present). Versioning offers a focus view, not a context view: what you get to see are moments in an object's evolution, hardly its whole history. A number of interesting research works introduce, at modelling level, ideas to help overcoming weaknesses inherent to the abovementioned solutions, like [13]. Nevertheless, GIS-inspired solutions have found a wide audience among archaeologists, with a recurrent trend to try and implement "4D" information systems about the history of a site [14] [15]. But GIS and CAD tools are basically concerned with distributing things in space. Consequently, **x** and **y** axes of the representation are requisitioned for spatial data – and time has to manage with what is left. Time geography may have started from the same observation – the **z** axis is here used to superimpose on a basic **x,y** cartography movements across a territory [16]. But the primary goal of time geography is to record and represent movements occurring in space over short periods of time. In our case artefacts do not "move" (or rarely…): they just change, and over long periods of time. Time geography is a promising concept - its adequacy to represent architectural changes remains questionable.

Widespread in information sciences, timelines, time bars, time charts are used to represent dates, periods, etc. A wide attention is notably put on the issue of visualising time-related phenomena in the field of information visualisation (see for instance [1], [10]). Such graphics do help reading a chronology, with events, durations, and possibly uncertainty assessments. A number of generic solutions have been developed in the past years, with some convincing results like [17]. But because time is represented in space – typically with a line or a spiral - spatial features tend to be scattered here

and there, causing visual discomfort rather than helping to understand spatial changes. The chronographs experience [9] taught us that although a timeline-like visualisation does greatly enhance reasoning about architectural changes, it is not well suited to spatial reasoning.

With computer implementations, interactive time sliders have been successfully introduced in many research and/or communication works [8]. As mentioned before, time sliders allow users to choose time slots they are interested in, and investigate the corresponding spatial configuration. But paradoxically, time sliders are mainly efficient in helping us to understand spatial changes. In fact, they are fundamentally a mean to browse through versions of objects using time as selection criteria, and in no way a context+focus visualisation. In a recent experiment we added to this time slider mechanism a visualisation of densities of changes: this context view however only focuses on time aspects – spatial changes remaining readable only one at a time.

As alternative, visual metaphors can be proposed that link spatial changes and a chronology – T.Ohta presents some nice concepts in [19]. In previous works, we developed a 3D metaphor called Infosphere, and a 2D metaphor called "ladybug race" that were equally "enjoyable". But neither the former (no durations, 3D navigation disconcerting for some users) nor the latter (no neighbourhood relations) did solve the problem. And so at the end of the day we are left with a number of solutions that correspond only in part to what we expect from a visualisation of dynamics of change.

The concentric time visualisation can be seen as coming to an arrangement between some of these solutions. It capitalizes on basic, well-known, well understood representations (cartography, timelines). It differs from them by introducing a specific, context+focus, graphic combination of spatial features and chronology. It is designed to help researchers reconsider their knowledge and the doubts that travel with it, and puts space and time on equal terms.

3 The Concentric Time Visualisation: Principles, Implementation

To start with, let us introduce some of the terms that we will use in this section:

- the **evolution** of an artefact is the time span separating its creation from its extinction (*i.e.* full physical removal, including of sub-structures, [9]);
- **morphological changes** imply transformations of the artefact's "shape",
- **recurrent changes** may concern upkeep, ownership, function, etc.,
- **contours** are simplified representations of an artefact's plan,
- **uncertainty** may concern the dating of events, the very existence of events, or the physical reality of an artefact. In all cases it is a consequence of the evidence we base on. It is represented in this experiment by a simplified lexical scale (known, uncertain, hypothetical).

Figures below present the visualisation's construction as a series of steps introducing spatial or temporal information - these steps should not be interpreted as "successive actions inside the construction procedure". The left part of figures illustrates the idea, the right part what it looks like when applied to our real case (for enhanced readability of the paper output, original contrast and line thicknesses of the computer output are accentuated).

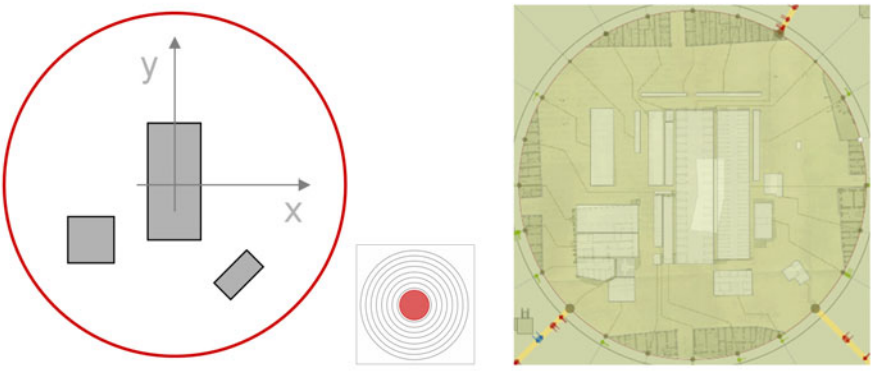


Fig. 1. A 2D map is displayed inside a red circle, showing simplified outlines of the artefacts under scrutiny. These contours are there to get an idea of where the artefacts were located, but they are not exact contours[3]. With no reasonable justification, an old map acts as a background.

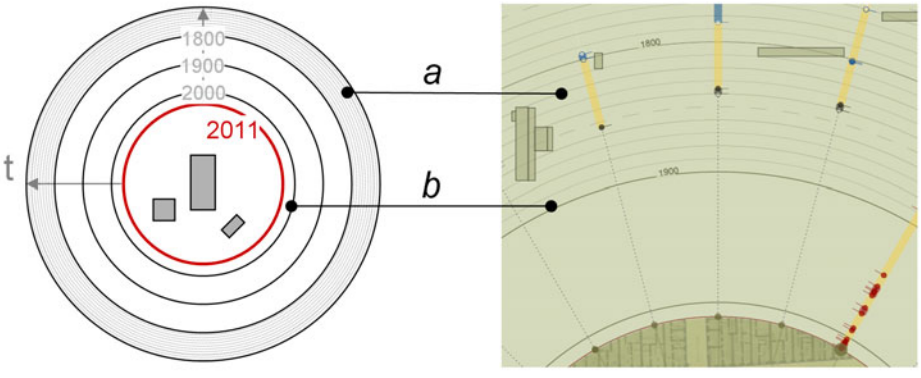


Fig. 2. Starting from the red circle (that stands for the current year - 2011), concentric circles represent a move towards the past. A circle is drawn for each decade (***a***) or century (***b***), except during the XXth century (this for readability purposes, considering very few changes occur then in our case study). The farer a circle is from the center, the older it is.

[3] There are several good reasons to this choice:
- A number of artefacts were built over another, older one – in other words at the same x,y position. Drawing contours one over the other would result in a good degree of visual pollution. Accordingly contours are represented in such a way as to minimise overlapping.
-If we were to represent a given version of an artefact on loading of the system, we would have to privilege one version of an artefact over another – and there is no reason to do so. Furthermore, if we were to privilege version 1 for artefact ***λ*** and version 3 for artefact ***δ***, we could end up with showing side by side edifices that never were there at the same moment. So that means we should rather select a given moment in history – well in that case we would just not see the whole set of contours since they never were all present at the same time.

In short, showing approximate contours enhances readability, and avoids spatial or temporal inconsistency.

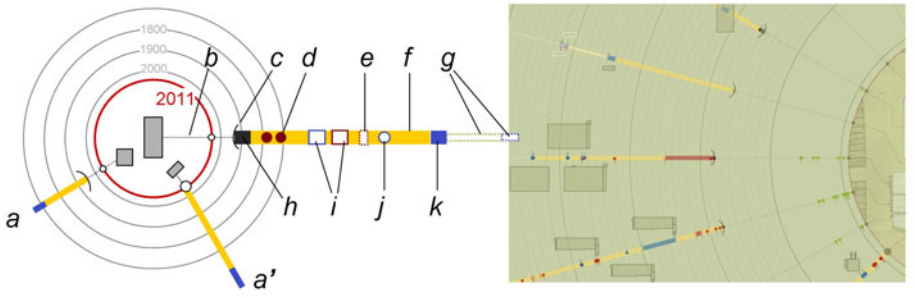


Fig. 3. Artefacts are connected to radial timelines (***a, a'***) that distributes events in time and recount changes through two graphic variables:
- Four colours : yellow (periods of stability, no known change, ***f***), blue (morphological change, including construction, ***k***), red (recurrent change, ***d***), grey (destruction, ***c***, meaning here that no visible trace is left above the ground).
- Shapes: differentiate long-lasting events (duration > year, represented by rectangles – ***l, e, k***) from punctual events (duration < a year, represented by a circle - ***d***, ***j***). When the dating of a period is uncertain, coloured rectangles as well as circles are filled in white, and only outlined in colour (***i, j***). When the dating of a period is hypothetical, the outlined is dashed (**e, g**). When the very existence of an artefact is uncertain, the size of coloured rectangles is modified (half-width, ***g***).
Visible on the right figure, an arc marks an artefact's destruction (***c***). It is withdrawn from the representation when zooming in, in order to lower the overall graphic weight. Timelines for artefacts that remain up to now are continued up to the central circle (***a'***). Capitalizing on the graphics' radial structure, dotted grey lines connect the timeline to a position on the central circle; white lines extend short timelines backwards in time (***b***).

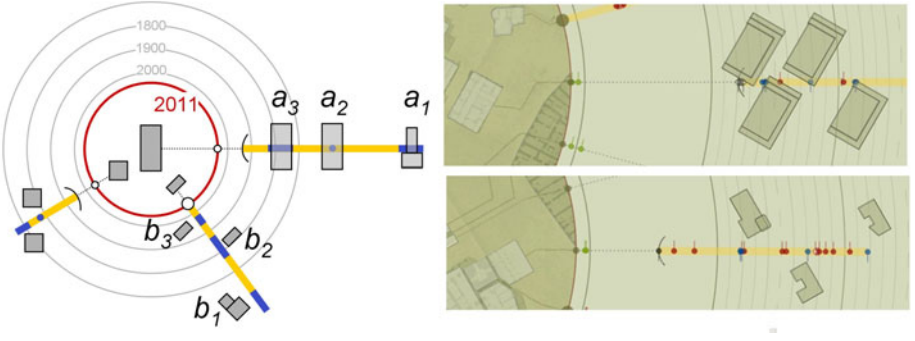


Fig. 4. All along the timeline, a new contour is drawn for each "blue" (morphological) transformation. The contour is drawn either on the timeline or above/below the timeline (a point we re-discuss in the implementation section).
Depending on what really happened, the actual geometry of the new contour may be left unchanged from version to version (***a*** - adding a new storey for instance does not change the artefact's contour). However, the visualisation acts also an interface, each contour is connected to queries that are specific to it – allowing users to retrieve info (basic data, 2D/3D content, bibliography, etc.) about what really happened (this point is re-discussed in the perspective section). The timelines radiate around the central circle, but the contour's orientation to the north is maintained whatever angle the timeline is drawn at.

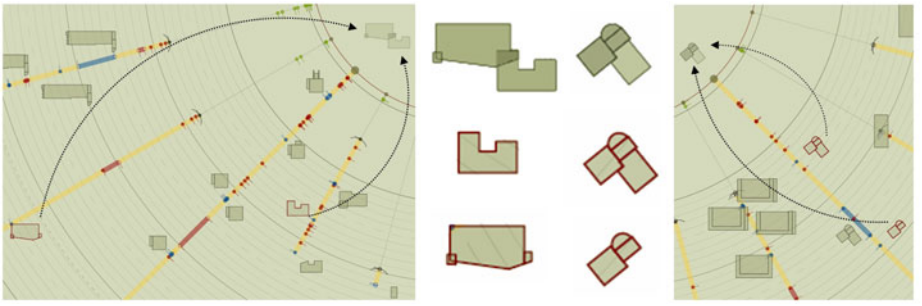


Fig. 5. Finally, in order to allow basic spatial analysis, each of the contours can be projected inside the central circle on user selection. Because contours are drawn with a level of transparency, this mechanism enables user-monitored comparisons of contours at two levels:
- Left, Neighbourhood analysis – in order to uncover the likelihood of event propagation - fires for instance, a plague at those periods.
- Right, Transformation analysis – shows the density of use of the ground over time, or helps spotting the period of emergence of a given component, and check on neighbouring artefacts whether "there is room for the added component" at that moment in history.
When contours are projected, the central circle's content is simplified (background image is hidden as well as simplified contours) so as to avoid visual overload.

The concentric time visualisation has been applied to the evolution of Krakow's Main Square (24 artefacts, 79 "contours", 391 "events" over 750 years). Although we already knew quite well this site, the visualisation did help us spot some interesting patterns - like an unexpected pattern of uncertainty on XIXth century destructions in the north-west corner. The following figures illustrate on two examples how the visualization may support reasoning tasks.

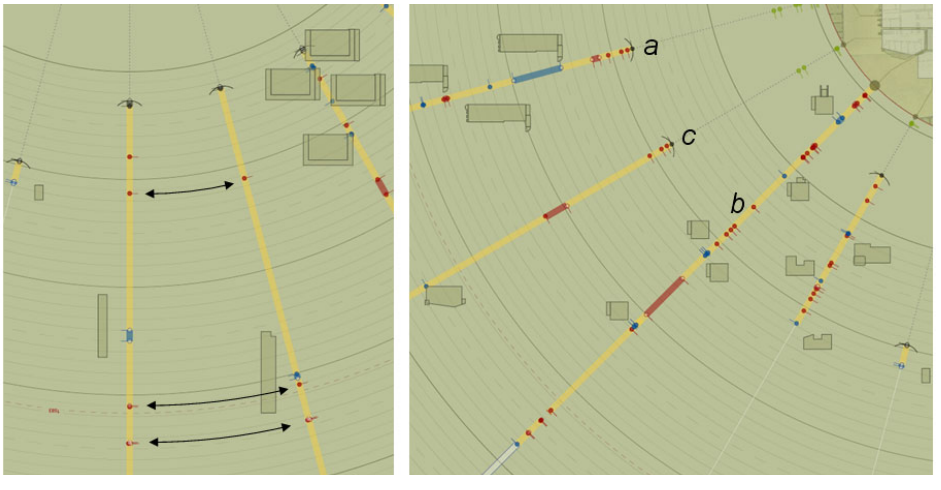


Fig. 6. Left, a focus view helps analysing causal relations between events and changes on neighbouring objects note parallel recurrent changes in both artefacts (red dots) independently of morphological changes. Right, a focus view allows comparisons of patterns between objects: compare densities of changes for (***a***) , (***b***) to this of (***c***) (a prison).

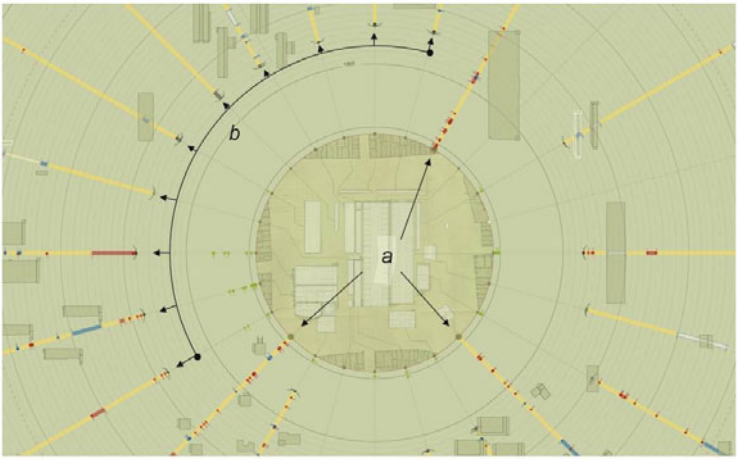


Fig. 7. Left, context view underlines features of the collection: for instance the proportion of remaining artefacts (***a***), 3 out of 24, and the impact of mid XIXth century destructions (***b***)

The implementation is a fairly primitive one, combining two main levels:

- Reusable components, composed of an RDBMS where information for each artefact are stored (dating, sources, description of each of the artefact's change) and of XML files (geometrical information for each version of an artefact. These components, developed in previous experiments [8][9], were here only updated with some recent archaeological evidence.
- The visualisation itself – an SVG file [18] produced on the fly4 for standard web browsers (Perl classes – architectural ontology – and modules developed in the abovementioned experiments), with user interactions monitored inside Javascript modules (zoom/Pan, show/hide, contour projection, queries, etc.).

Finally, the visualisation also acts as an interface: specific menus are available either over contours or over events that allow the querying of various data sets (RDBMS, XML files, as well as static 3D models corresponding to each contour).

4 Evaluation, Limitations, Perspectives

The evaluation was carried out with a group of eleven students in mechanical engineering (no background in architecture or historic architecture, no knowledge about the case study). Willingly, the session lasted less than 30 minutes, including the time

[4] A puzzling problem was how to position contours along the timeline (across, above, below? – see Fig4.). No systematic solution appeared as satisfactory. For each contour we specify manually, inside a javascript "configuration" file, an x,y move from its "natural" position (which is the center of a blue rectangle / circle). The visualisation is dynamically written with "natural" positions, and an onload jscript event moves the contours according to the configuration file. Contours can also be freely re-positioned by users (mouse dragging).

needed for us to present the visualisation's objective and graphic codes. The visualisation was projected on a white screen; with zoom levels fitted to the various questions. Testers were asked 14 questions distributed into 5 groups (space reading, chronology, uncertainty, comparing/analysing individuals, correlating space and time).

In the first three groups of questions we tested the visualisation's readability: testers were asked to decode the visualisation and to write down "how long does transformation t lasts", "which in this group of artefacts does not fit", etc. Facing questions of the two last groups, testers had to do some reasoning with questions like "identify pattern p", "can modification on artefact λ be related to events in the neighbourhood", etc.

Results show an error rate of 3%, mostly on readability questions. With such a low error rate, our questionnaire may be considered as not demanding enough. But answers were easy to give *provided the visualisation was easily and unambiguously understood*: and this is precisely what we were eager to verify. The evaluation does indicate the visualisation itself is almost self-evident, and moreover that some domain-specific notions (levels of credibility or uncertainties in particular) were correctly grabbed by full beginners in the field. We make no other claim about the evaluation. What has to be said is that the visualisation is a result of the interpretation of more than 400 bibliographic sources. Accordingly an evaluation that would allow comparing "reasoning from sources" and "reasoning with the visualisation" is far from being easy to build. But it is clear that we do not consider this evaluation as a definitive one. A number of issues remain open, for which future works will be needed:

- Case-study bias. The spatial layout of artefacts considered is our case study is well suited to the visualisation (artefacts kept close to one another for instance). A more complex spatial configuration, with more artefacts, with artefacts scattered in a wider area, or with more complex overlapping, may result in less convincing visual results. Tests on other case studies should be considered.

- Some additional features would definitely improve the visualisation's usability: independent zooming on spatial/temporal data, multi-resolution and partial zooming on timelines for short events, alternative levels of details for contours, etc. Our objective was not to come out with a ready-to-use system, but to test a concept: it is possible that this concept may be better implemented with a different technical set.

- In this experiment, only contours of artefacts are offered for users to carry out spatial analysis. Spatial analysis should be broadened to other 2D/3D characteristics.

- What the visualisation really does is nothing more than placing time and space side by side, and having them interact. Whether a deeper integration would bring better results remains to be established.

5 Conclusion

We introduce a context+focus visualisation called concentric time aimed at summarising the journey through time of groups of artefacts. The visualisation can be used for research purposes, but was evaluated with as ulterior motive testing its usability for a wider public, with possible museology applications. It was designed in order to combine inside a unique information space, spatial features and chronology, with as constraints uncertainty assessment, interface capabilities, and simplicity. The concentric time visualisation has been applied and evaluated on a case study, thanks to which we checked its support in carrying out reasoning tasks about architectural changes. Its

limitations are numerous: what were investigated here really are the possible benefits of the concept – no claim is made on a generic, ready to use, system. On the other hand, the concept appears as fairly generic: its usability beyond clarifying architectural changes could be investigated (for instance in geoinformation). Providing models and visual tools to handle dynamics of change remains today a hot research topic: we view our contribution as demonstrating that besides facing sometimes complex knowledge modelling challenges, researchers in the field also face the challenging complexity of simple visual thinking.

References

1. Spence, R.: Information vizualisation. Addison Wesley, ACM Press, Essex (2001)
2. Bertin, J.: Sémiologie graphique. EHESS, Paris (1998)
3. Card, S.K., Mackinlay, J.D., Schneiderman, B.: Information vizualisation: Using vision to think. Morgan Kaufmann, San Francisco (1999)
4. Sanders, L. (ed.): Models in Spatial Analysis London. ISTE, Newport Beach (2007)
5. Dürsteler, J.C.: Visualising Time (2006), `http://www.infovis.net/` (accessed February 2011)
6. Jensen, M.: Semantic Timeline Tools for History and Criticism. In: Digital Humanities, pp. 97–100 (June 2006)
7. Kienreich, W.: Information and Knowledge Visualisation: an oblique view. Mia Journal, 7–16 (2006)
8. Dudek, I., Blaise, J.–Y.: From artefact representation to information visualisation: Genesis of informative modelling. In: Butz, A., Fisher, B., Krüger, A., Olivier, P. (eds.) SG 2005. LNCS, vol. 3638, pp. 230–236. Springer, Heidelberg (2005)
9. Blaise, J.Y., Dudek, I.: Profiling artefact changes: a methodological proposal for the classification and visualisation of architectural transformations. In: Digital Heritage, Archeolingua, Budapest, pp. 349–356 (2008)
10. Tufte, E.R.: Visual explanations. Graphic Press, Cheshire (1997)
11. Maeda, J.: No simplicity without complexity. In: Schuller, G. (ed.) Designing Universal Knowledge, pp. 138–143. Lars Muller Publisher (2008)
12. Friendly, M.: Re-visions of Minard. Statistical Computing and Graphics Newsletter 11(1) (1999)
13. Rodier, X., Saligny, L.: Social features, Spatial features, Time features : An urban archaeological data model. In: Posluschny, A., Lambers, K., Hertog, I. (eds.) Proc. CAA (2007)
14. Johnson, I.: Mapping the fourth dimension: the TimeMap project. In: Dingwall, L., et al. (eds.) Archaeology in the Age of the Internet. British Archaeological Rep. Int. Series, vol. 750 (1999)
15. Van Ruymbeke, M., et al.: Development and use of a 4D GIS to support the conservation of the Calakmul site. In: Lasapona, R., Masini, N. (eds.) Proc. 1st International EARSeL Workshop, Aracne, pp. 333–338 (2008)
16. Yu, H., Shaw, S.-L.: Revisiting Hägerstrand's Time-Geographic Framework for Individual Activities in the Age of Instant Access. In: Proc. Research Symposium Salt Lake City, pp. 103–118 (2005)
17. Sabol, V., Scharl, A.: Visualizing Temporal-Semantic Relations in Dynamic Information Landscapes. In: AGILE 2008 Int. Conf. on Geographic Information Science (2008)
18. Rathert, N.A.: Knowledge visualisation using dynamic SVG Charts. In: Geroimenko, V., Chen, C. (eds.) Visualizing information Using SVG and X3D, pp. 245–254. Springer, Heidelberg (2005)
19. Ohta, T.: Diagram design course. In: Informational Diagram Collection, pp. 213–222. Pie Books, Tokyo (2009)

Analysis of Synergetic Aerodynamic Process by Means of Locally Asymptotic Estimation Derived from Telemetry Data

Victor F. Dailyudenko and Alexander A. Kalinovsky

United Institute of Informatics Problems NAS of Belarus,
Surganov St. 6, 220012, Minsk, Belarus
selforg@newman.bas-net.by

Abstract. Locally asymptotic estimation is derived from telemetry data processing by means of topological nonlinear method of temporal localization. Convergence for the function of topological instability at changing dimensionality is attained, and high reliability of diagnosis in a case of emergency caused by failure of equipment unit is proved. The essential reduction of computation time and required experimental data is also attained. Telemetry data are generated from computational modeling of the aerodynamic process within the restricted region of a spacecraft. The proposed method is shown to be useful for diagnosis of onboard equipment condition.

1 Introduction

The problem of timely detection of failures in onboard equipment of a spacecraft is a task of vital importance and plays a key role in successful arrangement of air flights and taking proper decisions in emergency situations. In this paper we show that this problem can be solved by telemetry signal processing using nonlinear dynamics algorithm analysing the attractor of investigated process. In general, similar approaches to exploration of geoscience and telemetry data have been widely applied during past decades [1-2] because those provide essential information that can not be obtained using traditional spectral-correlation methods.

But it is worth noting that computational complexity of nonlinear algorithms being applied for topological dynamics investigation makes these algorithms rather cumbersome from standpoint of their computer time and the quantity of required experimental data N [1, 3]. Therefore, the total time of observation and diagnosis process becomes rather long that may result in difficulties and even make impossible a prompt decision necessary in emergency situation. So, in this paper we develop the topological method based on temporal locality approach proposed in refs. [4-5] for overcoming such problems. In comparison with the most conventional methods of nonlinear analysis based on spatial localization [1-3], the developed method allows essential reduction of N and computation time as well as is insensible to growing m on these characteristics.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 642–652, 2011.

2 The Algorithm for Derivation of Locally Asymptotic Estimation of Topological Instability at Changing Dimensionality

In this section we show that the study of the topological structure of the attractor of the system can be reduced to the analysis of investigated time series (TS), the obtained asymptotic estimations of topological instability at increasing embedding dimension providing useful information about physical state of the process.

Let us consider a synergetic process described by a system of *d* nonlinear differential equations with *d* kinetic variables. The method of delayed coordinates (affirmed mathematically by Takens [6]) for reconstruction of phase trajectories forming an attractor R_T^m is given by [1-3, 6, 7]

$$\vec{x}_i^{(m)} = (\eta_i,\ \eta_{i+p},...,\eta_{i+(m-1)p}), \tag{1}$$

where $\eta(j\Delta t) = \eta_j$, $j = 1,2,...,N$ is a measured TS of a kinetic variable with a fixed time interval Δt, $\tau = p\Delta t$ is the delay time, p is an integer. The points $\vec{x}_i^{(m)} \subset R^m$, R^m is an Euclidean phase space with a dimension *m*. According to this method, the phase space of the attractor can be reconstructed by a TS of only one kinetic variable.

The common quantity of the attractor points is given by $L^{(p,m)} = N - p(m-1)$. In accordance with (1), phase trajectories can be represented as a superposition of *p* rarefied sequences $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_p$ shifted by one sample with respect to each other, those are defined as $\mathrm{X}_s = \{\vec{x}_{s+p(k-1)}^{(m)}\}_{k=1}^{L_s^{(p,m)}}$. These sequences are formed by *p* rarified TS $\Psi_1, \Psi_2, \ldots, \Psi_p$ obtained from the initial one, where $\Psi_s = \{\eta_{s+p(k-1)}\}_{k=1}^{N_s^{(p)}}$, $L_s^{(p,m)} = N_s^{(p)} - m + 1$. The quantity of samples involved by Ψ_s is given by

$$N_s^{(p)} = [\alpha_s]_{\mathrm{int}} + \mathrm{U}_+([\alpha_s]_{\mathrm{fract}}) \tag{2}$$

where $\alpha_s = \dfrac{N-s+1}{p}$; $[\alpha_s]_{\mathrm{int}}$ is an integer part derived by rounding α_s to the nearest integer towards zero; $[\alpha_s]_{\mathrm{fract}}$ is a fractional part of α_s being just a remainder after division; U_+ is a step function defined as follows

$$U_+(x) = \begin{cases} 1 & if \quad x > 0; \\ 0 & otherwise. \end{cases} \tag{3}$$

The formula (2) can be obtained by partition of the initial TS into separate segments of length *p* and counting initial samples of those.

At such consideration, the TS (excepting its initial samples), i.e. η_s, $\eta_{s+1},...,\eta_N$ is divided into separate segments such as $[\eta_s,\eta_{s+1},...,\eta_{s+p-1}]$, $[\eta_{s+p},\eta_{s+p+1},...,\eta_{s+2p-1}],\ldots,[\eta_{s+\bar{N}_s^{(p)}},\eta_{s+\bar{N}_s^{(p)}+1},...,\eta_N]$, where $\bar{N}_s^{(p)} = p(N_s^{(p)}-1)$, and $N-s+1$ is just the total number of samples involved by all segments. Calculation of the quantity of initial samples over all complete segments (i.e., with length p) yields the integer part in (2), and if the last segment is incomplete, then it provides the related fractional part. Evidently, $N_s^{(p)}$ experiences a decrease with growth of s (at least once during its changing).

As it was recently shown, rarefying on attractor points is reasonable for numerical simulation of fractal-topological analysis [1, 4, 5]. Otherwise, using points that are too close together in time leads to essential underestimates of the dimension, i.e. to aggravating accuracy of the topological analysis. So, we also implement temporal rarifying of phase trajectories for creating a subset of points with decorrelated components resulting in essentially random distribution in the embedding space. It is attained by the approach that is realized in the most convenient way, namely we use only one X_s for numerical experiments. That is constructed using the sequence, and rarifying is determined with p=2. Denoting components of Ψ_p for brevity as $\Psi_p = \{\xi_1,\xi_2,\ldots,\xi_{N_p^{(p)}}\}$, we obtain that terms of relative partition sequence $\{\mu_j^{(m)}\}$ constructed by means of segmentation of difference-quadratic TS are defined analogously [4, 5] as follows

$$\mu_j^{(m)} = \frac{\Delta\xi_{j+m}}{\bar{\sigma}_j^{(m)}} \tag{4}$$

where $\bar{\sigma}_j^{(m)} = \sum_{i=0}^{m-1}\Delta\xi_{j+i}^2$, $\Delta\xi_j = \xi_{j+1}-\xi_j$. Similarly to [4, 5], introduce the following measure of topological instability:

$$Z_\mu(m) = \sigma(\mu_j^{(m)}), \tag{5}$$

where $\sigma(\mu_j^{(m)})$ is the mean square variance, i.e. $\sigma(\mu_j^{(m)}) = \sqrt{\left\langle\left(\mu_j^{(m)} - \left\langle\mu_j^{(m)}\right\rangle\right)^2\right\rangle}$, the averaging is made over R_T^m points. For estimating the relative variance on $\{\mu_j^{(m)}\}$, introduce the following normalized instability function:

$$\breve{Z}_\mu(m) = \frac{Z_\mu(m)}{\left\langle\mu_j^{(m)}\right\rangle} = \left(\frac{\left\langle(\mu_j^{(m)})^2\right\rangle}{\left\langle\mu_j^{(m)}\right\rangle^2} - 1\right)^{\frac{1}{2}}. \tag{6}$$

Achievement of topological stabilization at sequential increase of embedding dimension of the attractor corresponds to the convergence of $\breve{Z}_\mu\ (m)$ to a constant level (or a dependence close to linear). The value of the phase space dimension, at which the convergence takes place, is taken as the minimum embedding dimension of the attractor m_0, this value is an important characteristic of the system.

The notion of topological instability is introduced analogously commonly used definition of instability of dynamical systems, i.e. when small deviation of kinetic variable can cause significant changes of its state [1], up to formation of new synergetic structures. The measure of topological instability (5) also shows the response of the attractor as a whole (determined by its topological structure [4, 5]) to minimal change of R_T^m embedding dimension. This response is estimated by relative changes of distances r_{ij}^m in R_T^m and reflects an average nonuniformity of $\{r_{ij}^m\}$ changes at enlarging dimension.

As it is shown in [4, 5], $\breve{Z}_\mu\ (m) \equiv 0$ in a case of ideal topological stabilization (ITS), when completely uniform change of distances takes place at enlarging dimensionality, and $\breve{Z}_\mu$ is just the asymptotic estimation of deviation from ITS. At the same time, normalization in a form of (6) reflects statistical indeterminacy at enlarging *m* because both numerator and denominator in (6) tend to zero at $m \to \infty$. Nevertheless, in this paper we prove numerically that this dependence provides useful information in a case of telemetry signal processing. Application of the derived $\breve{Z}_\mu\ (m)$ allows us to reduce the study of the topological structure of the attractor of the system to the analysis of the investigated TS.

3 The Scheme of Numerical Simulations of Aerodynamic Measurement for Onboard Equipment

The basic scheme of aerodynamic processes modeling is shown in Fig.1 (a). It is just a typical example which allows us to create the working model of necessary air born craft devices [8-10]. The main elements of the onboard equipment are the thermal control system, a heating unit and research unit. We consider, that in order to carry out properly conducted experiments on board, we must maintain the necessary conditions for the adjacent layer of air. Therefore, the model telemetry data were recorded near the middle of the six faces of the research unit.

Modeling of the aerodynamic flow is performed by solving the Navier-Stokes equations, which in Cartesian-tensor notation [10] are as follows:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = s_m \\ &\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j u_i - \tau_{ij}) = -\frac{\partial p}{\partial x_j} + s_i \end{aligned} \tag{7}$$

where t is time, x_i – Cartesian coordinate (i=1,2,3, as we solve 3D task), u_i – absolute fluid velocity component in direction x_i, p – piezometric pressure, where $p = p_s - \rho_0 g_m x_m$ (p_s is a static pressure, ρ_0 is reference density, g_m are gravitational acceleration components and the x_m are coordinates related to a datum where ρ_0 is defined), ρ - density, τ_{ij} - stress tensor components, s_i - momentum source components.

Since the system does not seem to have big flow rates and it has a little limited volume, we solved the problem in the case of laminar flow. In the case of laminar flow, stress tensor is described by the following expression:

$$\tau_{ij} = 2\mu s_{ij} - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{8}$$

where μ is a molecular dynamic viscosity of the fluid. The rate of strain tensor is given by:

$$s_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{9}$$

To account for heat transfer, we used the "Segregate Fluid Temperature Model" of thermal flow. In this case, the energy equation can be represented in the following integral form:

$$\frac{d}{dt}\int_V \rho E dV + \oint_A (\rho H(\vec{v} - \vec{v}_g) + \vec{v}_g p)\cdot d\vec{a} = -\oint_A \vec{q}''\cdot d\vec{a} + \oint_A \vec{T}\cdot d\vec{a} + \oint_V \vec{f}\cdot\vec{v}dV + \oint_V s dV \tag{10}$$

where E is the total energy, H is the total enthalpy, $\vec{q}''$ is the heat flux vector, $\vec{T}$ is the viscous stress tensor, $\vec{f}$ is the body force vector representing the sum of all body forces, $\vec{v}$ is the velocity vector, and $\vec{v}_g$ is the grid velocity vector (in our model is zero valued), s contributes additional energy source terms.

To construct the finite-element grid (Fig.1 (b)), we used the polyhedral cells, as they allow us to create high-quality geometry with a small number of cells (this allows us to vary widely elementary volume near the small and large borders). We created a double layer of cells near the boundary of the type "wall" for better accuracy of the flow effects near boundaries.

The main boundary conditions used to solve the problem are as follows:

- the rate of flow produced by a fan - 2 (m/s);
- temperature at the boundary thermoregulatory system - 223K;
- temperature at the boundary of the heating element - 333K;
- temperature initialization - 293K.

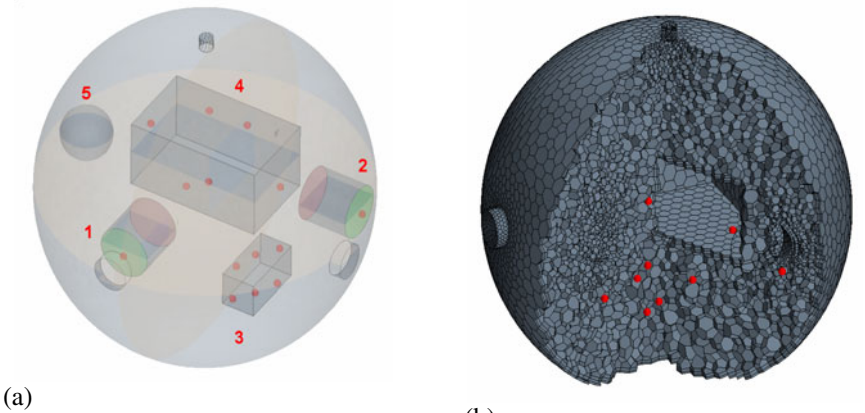


Fig. 1. (a) The general location scheme of constructive elements in the spacecraft: 1,2 – fans, 3 – research unit, 4 - device creates an additional heating (heating element), 5 – thermoregulatory system. Red dots correspond to the location of the telemetry data sensor; (b) Generated finite-element computational mesh. Double layer of cells on walls is present

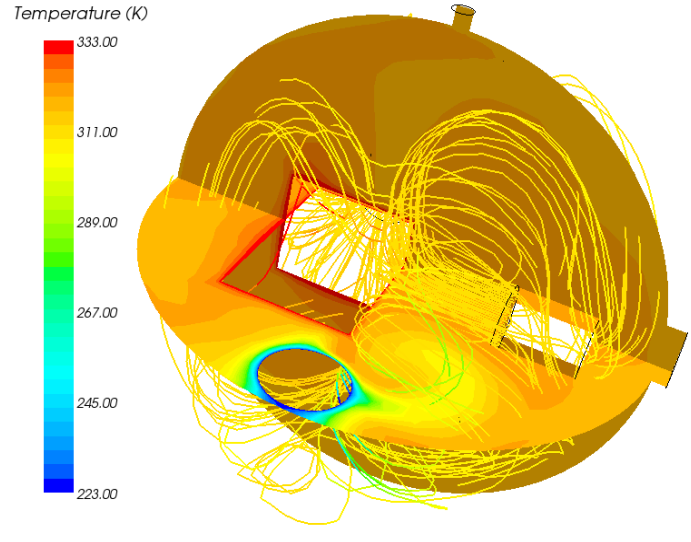


Fig. 2. The result of simulations: distribution of the temperature inside the working space spacecraft and streamlines for the velocity distribution

In order to correctly solve the problem of telemetry data calculation, the time-dependent regime of simulation is required (transient solver). However, this method of calculation significantly slows down the calculation. Therefore, to reduce the

computation time, we previously hold steady calculation problem, which later was used as the initialization data for transient mode of the solver. As virtual sensors, we used the six points near the boundary of the research module (Fig.1 (a)). The measured parameters were: pressure (P), velocity magnitude (Vm) and temperature (T) in a form of time series (TS).

Using a computational model, we implemented a virtual simulated emergency situations on board spacecraft when failure of one of the fans took place. For each mode of operation of internal spacecraft equipment one can observe characteristic distribution temperature field and the characteristic structure of the streamlines (Fig.2) of the velocity field. The difference of states was also observed on the test telemetry data received from a virtual sensor. We used these features for the subsequent processing of the telemetry data, thereby solving the problem of classifying the state of the spacecraft.

4 Numerical Simulations with Telemetry Data

In the calculation of topological dependencies, the following additional normalization (for unification of scale magnitude) is used:

$$\breve{Y}(m) = \frac{\breve{Z}_{\mu}(m)}{\breve{Z}_{\mu}(1)}. \tag{11}$$

It should be noted that the obtained topological relationships (their form, extreme points, average characteristics of convergence and dispersion estimates) are important characteristics of the state of the investigated nonlinear process, these estimates describe its attractor and can not be obtained with traditional linear methods, nor only from visual estimation of TS. The integer values of convergence directly characterize the complexity of the process under investigation. The usefulness and advantage of obtained dependencies is shown below on the basis of telemetric data obtained through numerical simulations of measurement of the air pressure in the confined space of a satellite for various modes of operation of onboard equipment. The process of numerical generation of pressure TS is implemented using the software for modeling of aerodynamic processes on a base of methodology described in section 2, common length of generated TS $\Im$ is 3000 samples, the first half of the string $\Im_1$ corresponds to the normal work of the two fans, while the second half $\Im_2$ describes the process in a case of failure of the one.

The calculated dependencies (11) are presented in Figs 3-4 (for normal operation of the two fans and in the case of a failure of one of them), they show the possibility of achieving convergence of the computational process at $m \geq m_0$ (similarly to study of fractal structure of a multidimensional attractor [1-3]). At the same time, the obtained topological dependencies differ significantly for these two cases, and in the second case, one can observe a minor sensitivity of the form of the topological curve to preliminary difference signal processing, i.e. when we take the difference TS $\Phi_{\mathrm{p}} = \{\varsigma_1, \varsigma_2, \ldots, \varsigma_{\mathrm{N}_{\mathrm{p}}^{(\mathrm{p})}-1}\}$, where $\varsigma_{\mathrm{i}} = \xi_{\mathrm{i}} - \xi_{\mathrm{i}+1}$, instead of initial TS Ψ_p. This fact

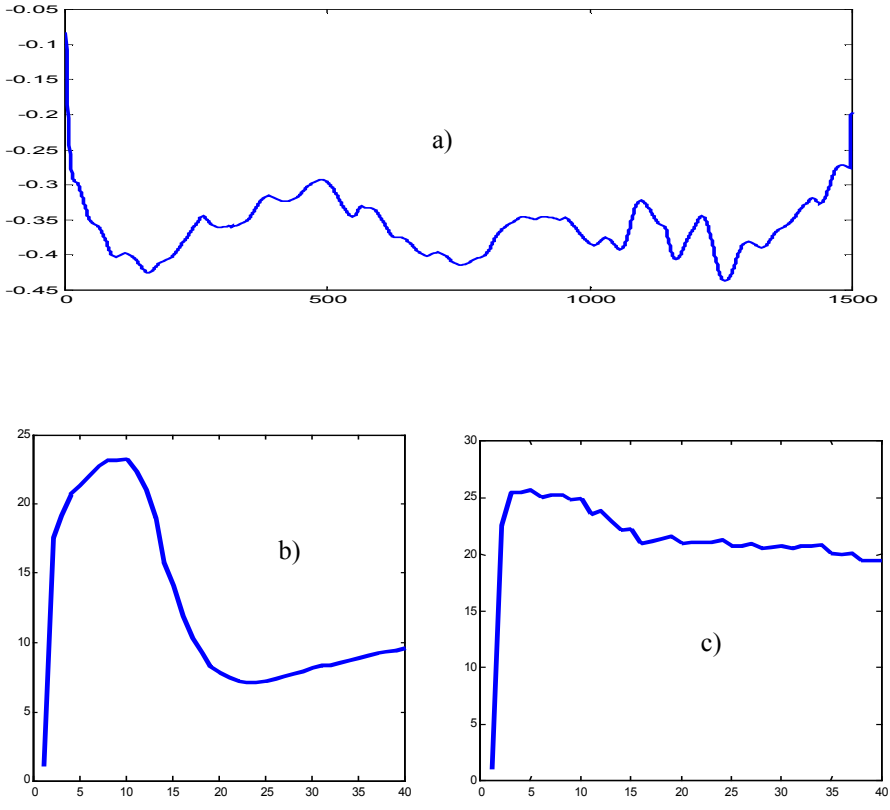


Fig. 3. Original time series of pressure describing aerodynamic processes (a) and normalized function of topological instability (b, c) vs phase space dimension obtained during normal operation of the two fans. The use of pre-difference signal processing allows us to obtain a smoother curve (c).

allows us to obtain additional information about the features of physical condition of the interior of the aircraft and provides an opportunity to implement timely detection of failure of the onboard equipment. The obtained numerical dependencies allow us to make conclusion that the simultaneous operation of two fans entails more complex aerodynamic process and formation of complex self-organization structures is possible that reflects in convergence at much larger values of the phase space dimension. This is in good coincidence with results of numerical simulations in section 2 where vortex structures in Fig 2 are evidently seen. The results for the modeling of measurement at other points in space inside the area of the aircraft are also obtained, they have similar form and confirm our conclusions.

Very urgent task for the rapid diagnosis of the functioning of onboard equipment in an automated way of data analysis by means of obtaining significant tags that would allow to implement the reliable and quick diagnosis, especially in cases of emergency. In this work, we propose for this purpose the value, which reflects the complexity of

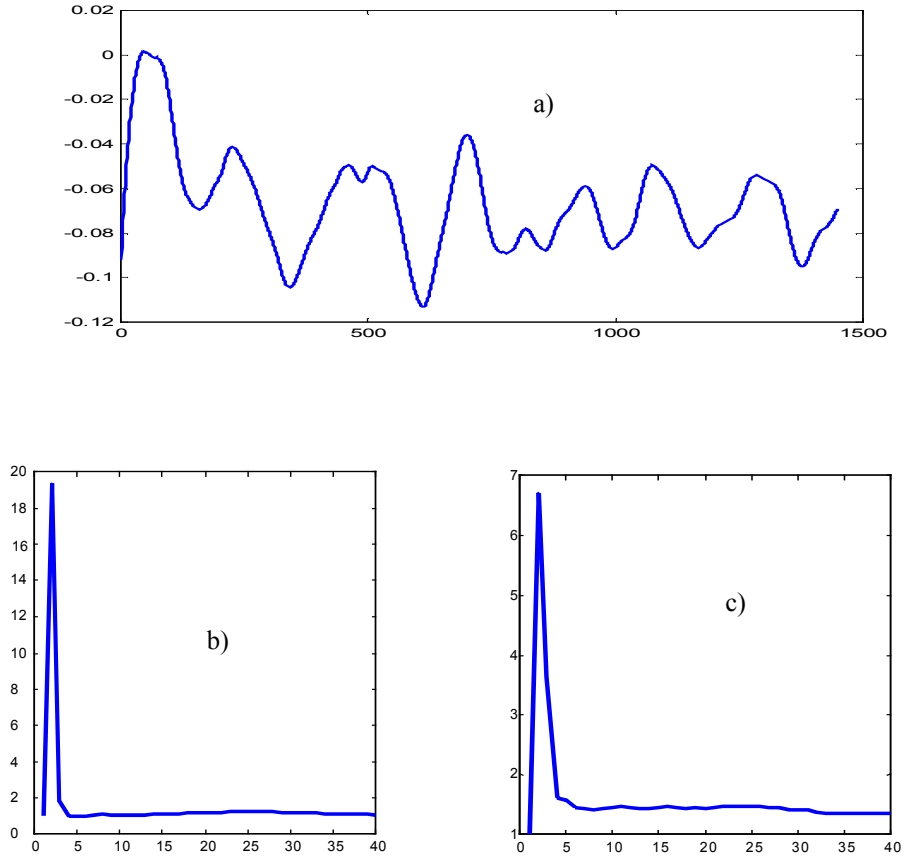


Fig. 4. Original time series generated in modeling of air flows (a) and normalized function of topological instability (b, c) obtained in the case of failure of one of the fans. The use of pre-difference signal processing are considerably DO NOT alter the shape of a topological curve (a), but reduces the level of the maximum. The level of convergence is much less than during normal operation of the two fans. The process is more periodic (a) and involves less complex dynamics.

the process and the size of the topological instability region, it is just the width $W_{\tilde{Y}}$ of the topological curve (Figs 3-4) obtained at the level $\frac{1}{2} max_m \{\tilde{Y}(m)\}$. Evidently, $W_{\tilde{Y}}$ determines also the level of dimension of phase space at which the convergence of the computational process to a quasi-linear dependence occurs.

In order to confirm the validation of the derived estimation $W_{\tilde{Y}}$, related calculations of it were carried out using topological curves obtained numerically from the following segments of the above TS $\Im$: $i = i_0 + 1$, $i_0 + 2$, ..., $i_0 + \Delta N - 1$, where i_0 = (s-1)dm. In this paper, we still assume ΔN = 1500, dm = 500, the parameter s

giving the shift of the initial segment of TS took successively the values 1, 2, 3, 4. The results of the numerical simulation yielded the following values: 15, 15, 15, 1, which shows a fairly high accuracy of estimation of the change in the state of the process in a case of failure of one of the fans.

5 Conclusions

Thus, in this work we show that the topological method based on temporal locality approach really provides a reliable estimation of the state of air born engineering that yields an opportunity to take a true decision in a case of emergency. This is implemented by simulation of aerodynamic processes on a base of physical features of air flows within a restricted region of a spacecraft. The advantage in computer time follows from the fact that the applied method includes minimal quantity of computer operations due to temporal localization and requires short TS (in our experiments N=1500, while realization of Grassberger – Procaccia algorithm (GPA) requires N=12000 ÷ 20000 in dependence on phase space dimension for proper calculation of probability dependencies characterizing a fractal structure [7]). For the algorithm developed in this paper, required quantity of experimental data N is constant for any $m \in [1, m_{max}]$ having linear growth only with enlarging m_{max}. At the same time, when using GPA the quantity N increases exponentially with growth of m region (see [1, 7] and references therein).

It should be also noted that the proposed algorithm is more universal because it is suitable for both TS forming fractal structure and for that having essentially non-fractal features that is very important in a case of real signal processing. Again, the method includes difference implementation in its structure that makes it useful in presence of noise because such difference operations provide decrease of noise level (both constant and periodic structure of noise influence). These advantages make this algorithm applicable for analysis of different technical systems including systems of thermal regulation whose modeling is considered in this paper.

References

1. Kruhl, J.H. (ed.): Fractal and dynamic systems in geoscience. Springer, Heidelberg (1994)
2. Volkov, S.A.: Stochastic Model of Time-Base Signal Errors in a Decametric Communication Channel, Allowing for their Fractal Properties. Telecommunications and Radio Enginering 68, 83–91 (2009)
3. Grassberger, P., Procaccia, I.: Characterization of strange attractors. Phys. Rev. Lett. 50, 346–349 (1983)
4. Dailyudenko, V.F.: Nonlinear time series processing by means of ideal topological stabilization analysis and scaling properties investigation. In: Proc. of the SPIE's Conf. on Applications and Science of Computational Intelligence II, Orlando, Florida, USA, vol. 3722, pp. 108–119 (April 1999)
5. Dailyudenko, V.F.: Topological considerations of an attractor based on temporal locality along its phase trajectories. Chaos, Solitons and Fractals 37, 876–893 (2008)
6. Takens, F.: Detecting strange attractors in turbulence. In: Dynamical Systems and Turbulence. Lecture Notes in Math., vol. 898, pp. 366–381. Springer, Berlin (1981)
7. Casdagli, M.: Nonlinear prediction of chaotic time series. Physica D 35, 335–356 (1989)

8. Vasiliev, V.V.: Complex thermal fields simulation on board the “foton” spacecraft. In: Proceeding 55th International Astronautical Congress, Vancouver, Canada, October 4-8 (2004), IAC-04-J.P.03
9. Abrashkin, V.I., et al.: Thermal Fields Modelling with Allowance of Microaccelerations at the ‘FOTON’ Spacecraft. In: Proceedings 52nd International Astronautical Congress, Toulouse, France (October 2001), IAF-2001-J.5.07
10. Warsi, Z.V.A.: Conservation form of the Navier-Stokes equations in general nonsteady coordinates. AIAA Journal 19, 240–242 (1981)

Landmark Detection for Autonomous Spacecraft Landing on Mars

Ugo Galassi

Università Amedeo Avogadro, Italy
galassi@mfn.unipmn.it

Abstract. In this paper we present a vision-based algorithm for estimating the absolute position of a *lander*, during the descent phase of a planetary exploration mission. A precise position estimation is required in order to avoid obstacles or to get close to scientifically interesting areas assessed on the basis of orbiter images.

Lander position is estimated with respect to visual landmarks on planetary surface. The core of the algorithm is a novel technique for identifying candidate landmarks using a local homogeneity analysis. This analysis can identify fast homogeneity changes or wide-range texture similarities and exhibits a computational cost that is invariant with respect to the size of the window where the measure is computed. Moreover, homogeneity analysis offers a way for simultaneously taking into account heterogeneous features.

The second contribution of this article is a general framework for position estimation based on landmarks, encoded by means of SURF descriptors. The relevance of the extracted features is increased in order to reduce the number of superfluous keypoints.

1 Introduction

Increasing the level of spacecrafts or robots autonomy is essential for broadening the reach of solar system exploration. A key motivation for developing autonomous navigation systems based on computer vision is that communication latency and bandwidth limitations severely constrain the ability of humans to control robots remotely.

A mission phase where autonomy is particularly important is Entry, Descent and Landing (EDL). In EDL the reaction time for choosing the right landing point is under a minute; then any interaction for an operator located on the earth is impossible.

Moreover, this phase may be particularly difficult because the scientifically interesting sites, target of the mission, are frequently located near craters, ridges, fissures, and other craggy geological formations. Then, to ensure a safe landing, the lander must autonomously identify its absolute position and the target landing site in order to avoid hazardous terrain.

Due to the lack of infrastructure such as global positioning system (GPS), past robotic lander missions relayed on traditional devices such as inertial measurement unit (IMU) and Doppler radar. The problem is that such equipments are quite imprecise (the error may be of several kilometers), due to the combined effect of noise, biases, and errors in initialization[4].

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 653–662, 2011.

For these reasons, most major space agencies agree on the need of developing autonomous landing systems heavily relying on vision. Known landmarks over the surface, such as craters, can be used, in vision-guided navigation systems, in order to estimate lander position. In [1] an approach is proposed where the landmarks are *craters*. The limitation is that there are sites where craters are not present. In such cases, a more general landmarks type is required, such as SURF keypoints [9]. SURF belong to the family of *local features*. All that is required to generate these features is a unique image texture, while, differently from crater detection, a model of terrain shape is not required. SURF keypoints can be reliably used to match images of different scale and orientation.

A pinpoint landing system, which uses local features as mapped landmarks to estimate the spacecraft global position, is proposed in [11]. A detector converts points of interest, in the descent image and in the geo-referenced orbiter image, into feature vectors (called *descriptors*), which later on are matched to each other. When the projections of at least four points with known global coordinates are detected in an image, an estimate of the camera pose can be computed. A typical problem, affecting many methods for landmarks identification, is the large number of spurious keypoints extracted, due to noise and false discontinuities. This dramatically increases the computational complexity frequently trespassing the resources provided by the on board computer.

The novelty presented in this paper is a *fast algorithm* for pinpoint landing that uses SURF keypoints as landmarks descriptors. Before SURF extraction, we perform a terrain analysis in order to identify strong discontinuities (corresponding to craters, rock, rim, etc.) and we limit SURF extraction to image regions corresponding to these visual landmarks. Due to this innovative approach, the number of descriptors that need to be extracted, stored and matched, dramatically decreases, significantly reducing the computational complexity. The visual landmarks selection step is based on a novel method for a multi-features analysis of the local homogeneity. This method computes the homogeneity of a specific feature over a local window and exhibits a computational complexity that is invariant with respect to the window size.

Keeping the computational complexity low is crucial because time constraints are a dominant requirement for EDL algorithms. In fact, the temporal window available for spacecraft localization during the parachute stage (between the heat shield jettison and powered descent), is limited to about 50 seconds. Although some algorithms can be implemented on specialized hardware, such as FPGAs [3], meeting such temporal requirements is made more difficult owing to the low computational power available on the on-board computers. The proposed algorithm is fast enough to cope with those constraints and has been proved to be effective with many different terrain conditions and sun illuminations. The algorithm has been evaluated using images taken by a robotic arm simulating the landing process on an accurate relief map of Mars.

2 Absolute Position Estimation Algorithm

Using SURF keypoints as landmarks, makes our algorithm applicable also for missions where specific landmarks like craters are not present in the landing area. The key idea exploited to tame the complexity consists in avoiding the extraction of descriptors in the whole image. For this reason, an operator for landmarks selection is applied to mask

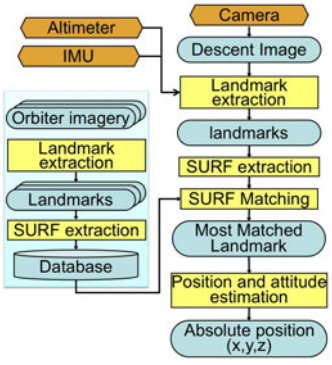


Fig. 1. Overall system architecture for position estimation

the image regions not corresponding to real landmarks. Visual landmarks correspond to discontinuities in the terrain surface. It is important to identify discontinuities in images that correspond to real changes in terrain morphology, while ignoring the others. False discontinuities can be due to shadows (projected by rocks, craters or the lander itself) or noise introduced in the image acquisition process. Then, the landmark selection is fundamental for reducing the vulnerability of SURF to false discontinuities. Moreover, reducing the number of extracted keypoints significantly decreases the memory requirements and the computing time. The overall architecture of the positioning system is presented in Figure 1. It consists of one off-line and one on-line part. In the off-line part, the 2D image of the foreseen landing area is obtained from the orbiter imagery. Visual landmarks are then extracted in the image, using the approach that will be presented in Section 2.1. A set of SURF signatures is defined for each of the extracted landmarks. The initial 2D position of landmarks, their SURF signature, and their 3D absolute co-ordinates on the surface are found in a database (keypoints database) stored in the memory of the lander. During the descent phase the algorithm inputs are: the keypoints database, the current image from the on-board camera (descent-image), and an estimate of the lander altitude and attitude. The algorithm performs the following steps:

(1) Descent-landmarks extraction and rectification: surface landmarks captured by the spacecraft camera are detected. The descent-image is rectified to the scale of the orbital image using a scale adjustment operator, which uses the altimeter information. The use of an altimeter is not essential but clearly improves the quality of persistent extracted landmarks.
(2) Potential candidate features extraction: a set of SURF features are extracted for each candidate landmark. This set defines the signature for the corresponding landmark.
(3) Landmarks retrieval: The signatures of landmarks present in the descent image are compared with pre-computed signatures stored in the database. A match is considered correct if the distance between the signatures is smaller than a given threshold.
(4) Spacecraft Position Estimation: The spacecraft position is estimated using modern-POSIT approach [7], given a set of matches between the landing image and the geo-referenced image.

Determining the 6 degrees of freedom pose (position and attitude) of a calibrated camera from known correspondence between 3D points and their 2D image counterparts is a classical problem in computer vision. It is known as the n-point pose problem or pose estimation. The solution of this well known problem is beyond the purpose of this paper and will not be discussed. For a detailed explanation of how to estimate the spacecraft position from a set of known landmarks, see [1,6,11].

2.1 Landmarks Extraction

The operator for landmarks extraction is based on a multi-cue approach. Firstly, a set $F = \{f_1 \dots f_K\}$ of significant features from the image is extracted. The majority of Mars image databases contains grayscale images, therefore the choosen features are magnitude and direction of the gradient, and pixel intesity. For each feature we compute its local homogeneity H_f using a local approach that will be described in Sec. 3. Using this approach we obtain a set of H-values, one for each feature.

The different H-values are then combined using their absolute weighted mean:

$$H = \frac{\sum w^{f_k} |H^{f_k}|}{K} \tag{1}$$

where $1 \leq k \leq K$ is the feature index and w^{f_k} is the associated weight. Negative weights can be used to reduce the relevance of misleading discontinuities, like shadows. In order to reduce the contribution of shadow areas we compute the homogeneity of pixel intesity both on the whole image and in the darker areas only. The H values for shadow areas will have a negative weight in Eq. (1) reducing the relevance of edges due to shadows.

The image is then segmented, according to the homogeneity values, using a pyramid based approach [2,10]. A schematic description of the process is given in Fig. 2.

3 Homogeneity Analysis

Different approaches to homogeneity analysis have been proposed by several authors [8,12,5]. Homogeneity is largely related to the local information extracted from an image and reflects how uniform a region is. It plays an important role in image segmentation since the result of a segmentation process is a set of several homogeneous regions. For each pixel, homogeneity analysis is performed considering a region of size $k \times k$ centered in pixel itself. The approaches presented in literature are typically time consuming and can be applied only by adopting small values for k. This means that only very short range regularities can be identified. In our method, the complexity is linear in the image size and remains constant for different values of k as we will demonstrate in the following.

The basic idea is that each pixel in the image can be considered as a particle p_c centered in a window $W_{k \times k}$ ($k = 2N + 1$) that is subject to different attractive forces, one for each other pixel in W. Let $p_i = (x_i, y_i) 1 \leq i \leq k$ be a pixel belonging to W and φ_i the value assumed by a generic feature Φ in that point. The force $\overrightarrow{F}_i$ applied

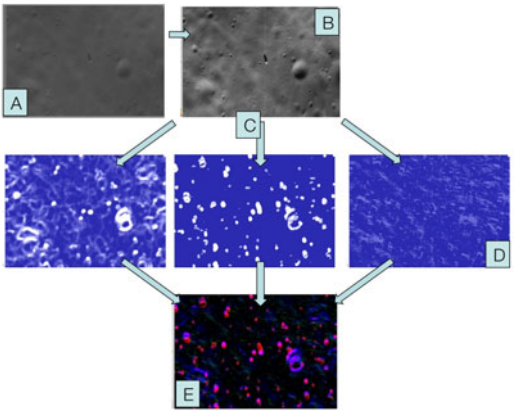


Fig. 2. Landmarks detection process: (A) original image (B) image enhanced; (B) feature extraction; (C) calculus of feature gradients using local homogeneity; (D) combining individual gradients according to Eq. (1)

to p_c by each pixel p_i is proportional to the difference between the values φ_i and φ_c, assuming that all pixels have the same mass.

Considering a Cartesian coordinate system centered in p_c, with axes parallel to image axes, the direction of the vector can be denoted with θ_i defined as the angle between the segment from p_c to p_i and the the x-axis. Then, for every pixel in the window it is possible to calculate the components (f_{x_i}, f_{y_i}) of vector $\overrightarrow{F}_i$ where:

$$f_{x_i} = (\varphi_i - \varphi_c)\cos(\theta_i) \tag{2}$$

$$f_{y_i} = (\varphi_i - \varphi_c)\sin(\theta_i) \tag{3}$$

Then, the force applied to p_c is computed as:

$$\overrightarrow{F}_c = \sum_{i=1}^{k^2} \overrightarrow{F}_i * m \tag{4}$$

being m the mass of each pixel. Assuming that each pixel has unitary mass, expression (4) can be rewritten as:

$$\overrightarrow{F}_c = \sum_{i=1}^{k^2} \overrightarrow{F}_i \tag{5}$$

The homogeneity H_c in p_c is therefore defined as the norm of $\overrightarrow{F}_c$:

$$H_c = \|\overrightarrow{F}_c\| \tag{6}$$

The resulting homogeneity values are normalized at the end of the analysis procedure, and are optionally normalized in the interval $[0, 255]$ in order to be represented as a grayscale image. A pixel located in a region that is homogeneous with respect to the

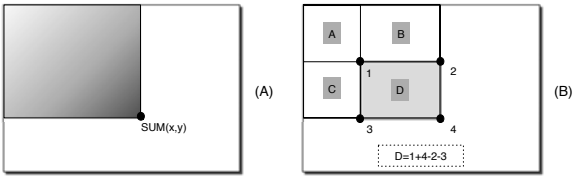


Fig. 3. The integral image can be used for computing multiple sums over arbitrary rectangular regions in constant time. It can be computed efficiently in a single pass over the image. We can define the algorithm recursively: $SUM(x,y) = val(x,y)+SUM(x-1,y)+SUM(x,y-1)-SUM(x-1,y-1)$ (a) The integral image, at location (x,y): $SUM(x,y) = \sum_{x' \leq x, y' \leq y}(x', y')$. (b) Using the integral image, the task of evaluating any rectangle can be accomplished in constant time with four array references: $sum(D) = sum(1) + sum(4) - sum(2) - sum(3)$.

feature Φ has a small value of H. On the other hand, a pixel located near a boundary between two regions characterized by different values of Φ shows a large value of H. A conceptually similar approach is presented in [8]. The problem in this kind of approach is the complexity that tends to be quadratic for large values of k.

A good approximation is obtained by subdividing the window W into a finite number Q of rectangular super-pixel $P_q (1 \leq q \leq Q)$ and by approximating the value φ_i over P_q with its mean value. Let A_q be the number of pixels in P_q, we can write:

$$\overline{\varphi_{P_q}} = \frac{1}{A_q} \sum_{i \in P_q} \varphi_i \tag{7}$$

As it will be explained further in Fig. 3, the mean value over a finite rectangular sub-region of the image can be computed in constant time, given the integral image of I. Under this approximation, the mass of each super-pixel P_q is given by A_q, whereas the angle θ_i is defined as the angle between the x-axis and the segment from p_c to p_m, being p_m the centroid of P_q. Different super-pixel morphologies can be adopted. In Fig. 4(a) it is presented the chosen one, where four super-pixels of equal size, partially overlapping, are defined. Using this approach $\theta_q = \frac{\pi}{4}(2q-1)$ being $q \in \{1,2,3,4\}$ the super-pixel number, in counterclockwise order. Based on these assumptions Eq. (4) can be rewritten as:

$$\overrightarrow{F}_c = \sum_{q \in \{1,2,3,4\}} \overrightarrow{F_{P_q}} * m \tag{8}$$

According to Eq. (2) and (3), we have $\overrightarrow{F}_{P_q} = (f_{x_q}, f_{y_q})$, where:

$$f_{x_q} = (\overline{\varphi_{P_q}} - \varphi_c)\cos(\theta_q) \tag{9}$$

$$f_{y_q} = (\overline{\varphi_{P_q}} - \varphi_c)\sin(\theta_q) \tag{10}$$

In the adopted subdivision all super-pixels have the same size and morphology. Therefore, we can ignore the mass value, setting $m = 1$ in Eq. (8).

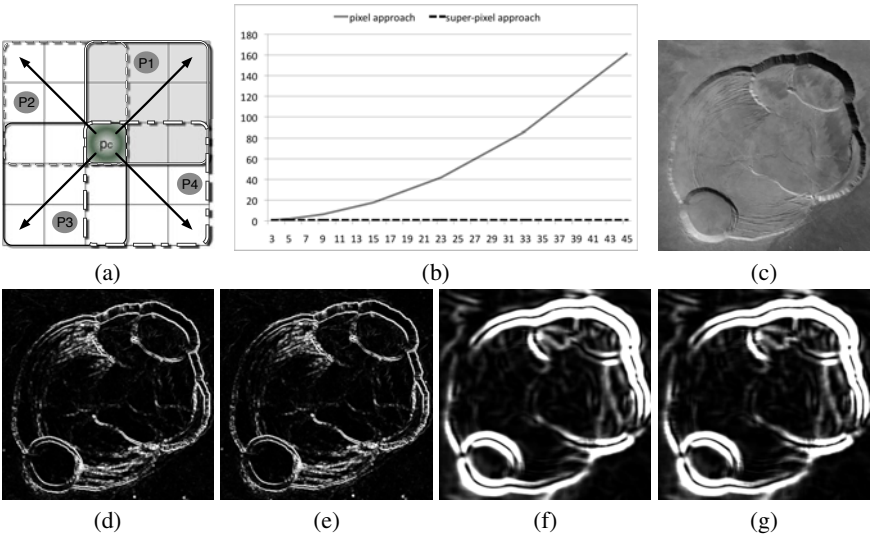


Fig. 4. Comparison of *pixel* versus *super-pixel* approach to homogeneity analysis. By increasing the window size, wide-range texture homogeneities are captured. (a) Super-pixels decomposition of window W. (b) CPU-time for computing homogeneity by using both approaches on the same image. The diagram reports seconds of CPU-time versus the local window size. (c) Original image. (d) *Pixel* approach [k=3]. (e) *Super-Pixel* approach [k=3]. (f) *Pixel* approach [k=21]. (g) *Super-Pixel*approach [k=21]. The use of *Super-Pixel* doesn't affect the quality of the results.

It is evident that the homogeneity value strongly depends on the window size. By using small windows, local noise on the value of φ is reflected on the value of H, producing dishomogeneous areas. Employing a larger window increases the smoothing effect, and H becomes less sensitive to noise. However, smoothing the local area could hide some abrupt changes in the local region, which should be preserved. We adopted a window of size $k = 5$ for computing homogeneity over gradient magnitude and pixel intensity, and a window of size $k = 25$ for gradient direction.

In order to clarify the validity of the proposed approximation, we report, in Fig. 4, a comparison of *pixel* versus *super-pixel* approach to homogeneity analysis. In the proposed example, the evaluated feature is the intensity value of each pixel. The results are approximatively identical in both approaches, making the use of *super-pixel* fully suitable for homogeneity analysis. Increasing the local window size allows to smooth out the changes in intensity due to the texture. Homogeneity is a kind of gradient and can be used also for edge detection purposes. It is evident that increasing the window size allows to capture the most significant edges in the image.

Figure 4(b) reports the CPU-time for homogeneity analysis in both approaches. Homogeneity over intensity has been computed on a grayscale image of size 800×1231. The local window has a size ranging from $k = 3$ to $k = 45$ with a step of 2. As reported in the graph, the computational time for the *super-pixel* approach remains constant ($\sim 0.36sec.$ on an intel core i7 @2.66Ghz) while the *pixel* approach ranges from $\sim 0.54sec.$ for $k = 3$ to $\sim 161.72sec.$ for $k = 45$.

4 Algorithm Evaluation

We tested our algorithm on a simulated landing scenario and on Mars images obtained by the Mars Global Surveyor (MGS) database. In both scenarios we evaluated its capability of recognizing the right target landing site as well as the reduction in the number of keypoints that are extracted.

4.1 Simulated Landing Scenario

The simulated landing scenario has been reconstructed inside a laboratory of *Thales Alenia Space*, by using a precise relief map and a variable Led color illumination system. A camera located on a robotic arm collected images simulating the visual input during a landing episode on Mars surface. The precision relief map is composed by nine panels of 300 x 300 mm, with a max height of 150 mm each. This relief has been realized by means of a particular 3D printer and is characterized by a precision that is better than 0.1 mm.

Two kinds of experiments have been performed. In the first one, two sets of 30 images corresponding to different landing trajectories (landing images) have been collected and compared to a database of 5 non-overlapping reference images. Each reference image corresponds to a different region in the map. The reference images have been taken without activating the LED color illumination system. On the contrary, the two sets of landing images have been acquired under different illumination conditions. For each landing image the system tries to identify the correct reference image. The task is performed with, and without, the landmark selection operator enabled. A landing image is assigned to the reference image having the maximum number of corresponding keypoints.

In Table 1(a) the main results for experiment 1 are summarized. The first row reports the global classification accuracy (*C.A.*). The second row reports the average number of extracted keypoints (k) for both approaches. Finally, it is reported the average value of the ratio between the number of true correspondences which have been found, and the total number of keypoints (*k-ratio*). The true correspondences have been evaluated by using the RANSAC algorithm [13]. It is evident that the drastic reduction in keypoints number doesn't affect the classification accuracy.

In the second experiment the robotic arm has performed a complex trajectory over the map simulating a spacecraft flying over the scenario. The same trajectory has been repeated 5 times and each time a sequence of 20 photographs has been taken. Camera position was always the same for the i-th photo of every sequence and the only difference was due to a change in the direction of the illumination. The illumination angles characterizing the 5 series are: 0°, 90°, 180°, 225°, 315°. We have compared all the images in the first series with the corresponding images in the other series and then computed the *k-ratio*.

The aim of the experiment is to analyse the number of spurious keypoints that are avoided by using the landmark extraction operator. The i-th photo of every sequence always presents an identical landscape and the only differences are due to the change in illumination. The strong increase in the *k-ratio* puts in evidence how the adoption of the operator for landmark selection principally affects only spurious keypoints. The

Table 1. Performances on simulated landing scenario

(a)	LE-disabled	LE-enabled
C.A.	0.94%	0.92%
$\overline{k}$	394	257
k-ratio	0.343	0.377

(b)	LE-disabled	LE-enabled
C.A.	0.96%	0.94%
$\overline{k}$	417	206
k-ratio	0.23	0.31

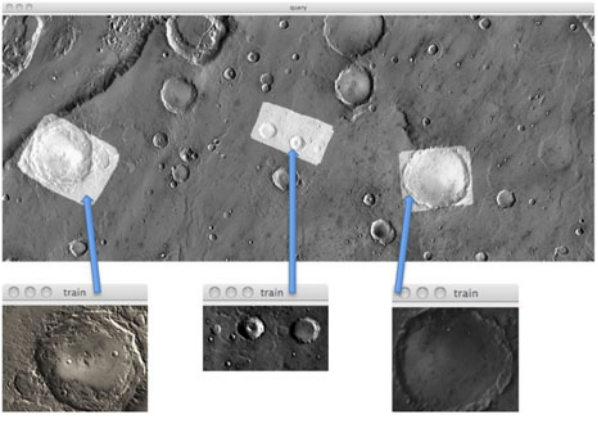


Fig. 5. Target sites are correctly identified in the reference image

results are summarized in Table 1(b). We presented also the average number of extracted keypoints and the classification accuracy with respect to the reference images used in the first experiment.

4.2 Analysis on Mars Images

We also evaluated the algorithm on real images using Mars Orbiter Camera (MOC) pictures, acquired by NASA's Mars Global Surveyor (MGS) orbiter. MOC consists of a narrow angle system that provides grayscale high resolution views of the planet's surface (1.5 to 12 meters/pixel), and a wide angle camera (200 to 300 meters/pixel). For each kind of camera we selected 10 pair of images. Each pair covers the same region but the two images have been acquired at different time. We used the first image of each pair as a reference frame. From the second image we extracted three different target sites (usually 10 or more time smaller than the reference frame). The system always succeeded in identifying the target sites in the reference frame, both with landmark selection operator enabled or disabled (Fig. 5). Enabling the landmark operator reduced the number of extracted (and stored!) keypoints of, approximately, 35% . The analysis time is reduced, indicatively, of 10%. Besides, the proposed approach has the advantage of making SURF keypoints less sensitive to the hessian value, i.e. the threshold used for accepting or rejecting new keypoints. With a too high threshold the number of extracted keypoints could increase dramatically. Reducing the image area on which they are extracted avoid that, in presence of a low thresold, their number can increase too much.

5 Conclusions

We presented a novel application for real-time landmark detection based on SURF descriptors and local homogeneity analysis. The presented approach limits the SURF extraction process to those discontinuities more likely corresponding to real changes in terrain morphology. Therefore, the number of SURF keypoints, which need to be extracted, stored and matched, is strongly reduced. The core of the application is a novel algorithm for local homogeneity analysis characterized by a computational complexity that is invariant with respect to the size of the local window used for computing it.

Acknowledgments. The present work has been supported by the STEPS Project (Project co-financed by EC Platform: POR FESR - 2007/2013). I would like to thanks *Thales Alenia Space* for allowing the use of their laboratory and the simulated landing facility. Special thanks to Piergiorgio Lanza for his invaluable support and his suggestions.

References

1. Ansar, A., Cheng, Y.: An analysis of spacecraft localization from descent image data for pinpoint landing on mars and other cratered bodies. Photogrammetric Engineering and Remote Sensing 71(10), 1197–1204 (2005)
2. Antonisse, H.J.: Image segmentation in pyramids. Computer Graphics and Image Processing 19(4), 367–383 (1982)
3. Barfoot, T.: Online visual motion estimation using fastslam with sift features. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3076–3082 (2005)
4. Braun, R., Manning, R.: Mars exploration entry, descent and landing challenges. In: Aerospace Conference IEEE (2006)
5. Chaabane, S.B., Sayadi, M., Fnaiech, F., Brassart, E.: Colour image segmentation using homogeneity method and data fusion techniques. In: EURASIP J. Adv. Sig. Proc., pp. 1–11 (2010)
6. David, P., Dementhon, D., Duraiswami, R., Samet, H.: Softposit: Simultaneous pose and correspondence determination. Int. J. Comput. Vision 59, 259–284 (2004)
7. DeMenthon, D.F., Davis, L.S.: Recognition and tracking of 3d objects by 1d search. In: DARPA Image Understanding Workshop, pp. 653–659 (1993)
8. Jing, F., Li, M.: jiang Zhang, H., Zhang, B.: Unsupervised image segmentation using local homogeneity analysis. In: Proc. IEEE Int. S. on Circuits and Systems, pp. 145–148 (2003)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
10. Marfil, R., Molina-Tanco, L., Bandera, A., Rodriguez, J.A., Sandoval, F.: Pyramid segmentation algorithms revisited. Pattern Recognition 39(8), 1430–1451 (2006)
11. Trawny, N., Mourikis, A.I., Roumeliotis, S.I., Johnson, A.E., Montgomery, J.F.: Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks. Journal of Field Robotics 24(5), 357–378 (2007)
12. Wang, H., Suter, D.: Color image segmentation using global information and local homogeneity. In: Sun, C., Talbot, H., Ourselin, S., Adriaansen, T. (eds.) DICTA, pp. 89–98. CSIRO Publishing (2003)
13. Wei, W., Jun, H., Yiping, T.: Image matching for geomorphic measurement based on sift and ransac methods. In: International Conference on Computer Science and Software Engineering, vol. 2, pp. 317–320 (2008)

Precise and Computationally Efficient Nonlinear Predictive Control Based on Neural Wiener Models

Maciej Ławryńczuk

Institute of Control and Computation Engineering, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
Tel.: +48 22 234-76-73
M.Lawrynczuk@ia.pw.edu.pl

Abstract. This paper describes a nonlinear Model Predictive Control (MPC) algorithm based on a neural Wiener model. The model is linearised on-line along the predicted trajectory. Thanks to linearisation, the algorithm is computationally efficient since the control policy is calculated on-line from a series of quadratic programming problems. For a nonlinear system for which the linear MPC approach is inefficient and the MPC algorithm with approximate linearisation is inaccurate, it is demonstrated that the described algorithm gives control quality practically the same as the MPC approach with on-line nonlinear optimisation.

Keywords: Process control, Model Predictive Control, Wiener systems, neural networks, optimisation, soft computing.

1 Introduction

In Model Predictive Control (MPC) algorithms a dynamic model of the process is used on-line to predict its future behavior and to optimise the future control policy [7,13]. In comparison with other control techniques, they have a few important advantages. First of all, constraints can be easily imposed on process inputs (manipulated variables) and outputs (controlled variables). Furthermore, they can be efficiently used for multivariable processes and for processes with difficult dynamic properties (e.g. with significant time-delays). In consequence, MPC algorithms have been successfully used for years in numerous advanced applications, ranging from chemical engineering to aerospace [12].

The simplest MPC algorithms use for prediction linear models. Unfortunately, for nonlinear systems such an approach may result is low quality of control. In such cases nonlinear MPC algorithms based on nonlinear models must be used [9,13]. Because neural models offer excellent approximation accuracy and have a moderate number of parameters, they can be efficiently used in nonlinear MPC [6,10,13]. The classical neural network is entirely a black-box model. It means that its structure has nothing to do with the technological nature of the process and its parameters have no physical interpretation. A viable alternative is a

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 663–672, 2011.

block-oriented model which consists of separate dynamic and steady-state parts. In particular, the neural Wiener model, in which the nonlinear neural steady-state part follows the linear dynamic part can be efficiently used for modelling, fault detection and control of technological processes, e.g. chemical reactors, heat exchangers, distillation columns, separation processes and evaporators [3].

This paper details a nonlinear MPC algorithm based on the neural Wiener model. The model is iteratively linearised on-line along the predicted trajectory. Unlike existing MPC approaches in which an inverse steady-state model is used to compensate for the nonlinear part of the Wiener model, e.g. [1,5,8,11], the presented algorithm does not need the inverse model. The algorithm is computationally efficient because the control policy is calculated on-line from a series of quadratic programming problems. It is demonstrated that the described algorithm gives control quality practically the same as the MPC approach with on-line nonlinear optimisation. The considered process is significantly nonlinear, the inverse of its steady-state part does not exist. The classical linear MPC algorithm is slow, the MPC algorithm with approximate linearisation [4] (in which the linear dynamic part is simply multiplied by a gain derived from the nonlinear steady-state part for the current operating point) gives unwanted oscillations.

2 Model Predictive Control Algorithms

In MPC algorithms [7,13] at each consecutive sampling instant k, $k = 0, 1, 2, \ldots$, a set of future control increments

$$\triangle \boldsymbol{u}(k) = [\triangle u(k|k)\ \triangle u(k+1|k) \ldots \triangle u(k+N_{\mathrm{u}}-1|k)]^{\mathrm{T}} \tag{1}$$

is calculated. It is assumed that $\triangle u(k+p|k) = 0$ for $p \geq N_{\mathrm{u}}$, where N_{u} is the control horizon. The objective is to minimise differences between the reference trajectory $y^{\mathrm{ref}}(k+p|k)$ and predicted values of the output $\hat{y}(k+p|k)$ over the prediction horizon $N \geq N_{\mathrm{u}}$. Constraints are usually imposed on input and output variables. Future control increments (1) are determined from the following MPC optimisation task (hard output constraints are used for simplicity)

$$\begin{aligned} &\min_{\triangle \boldsymbol{u}(k)} \left\{ \sum_{p=1}^{N} (y^{\mathrm{ref}}(k+p|k) - \hat{y}(k+p|k))^2 + \lambda \sum_{p=0}^{N_{\mathrm{u}}-1} (\triangle u(k+p|k))^2 \right\} \\ &\text{subject to} \\ &\quad u^{\min} \leq u(k+p|k) \leq u^{\max}, \quad p = 0, \ldots, N_{\mathrm{u}}-1 \\ &\quad -\triangle u^{\max} \leq \triangle u(k+p|k) \leq \triangle u^{\max}, \quad p = 0, \ldots, N_{\mathrm{u}}-1 \\ &\quad y^{\min} \leq \hat{y}(k+p|k) \leq y^{\max}, \quad p = 1, \ldots, N \end{aligned} \tag{2}$$

Only the first element of the determined sequence (1) is applied to the process, i.e. $u(k) = \triangle u(k|k) + u(k-1)$. At the next sampling instant, $k+1$, the output measurement is updated, the prediction is shifted one step forward and the whole procedure is repeated.

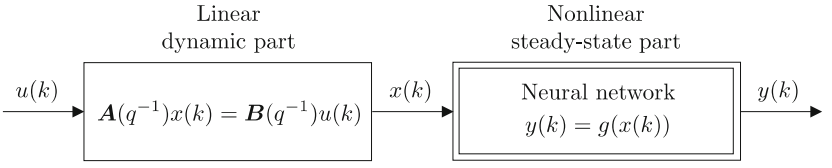


Fig. 1. The structure of the neural Wiener model

3 Neural Wiener Models

Predicted values of the output variable, $\hat{y}(k+p|k)$, over the prediction horizon are calculated using the neural Wiener model depicted in Fig. 1. It consists of a linear dynamic part in series with a nonlinear steady-state part, $x(k)$ denotes an auxiliary signal. The linear part is described by the difference equation

$$\boldsymbol{A}(q^{-1})x(k) = \boldsymbol{B}(q^{-1})u(k) \tag{3}$$

where polynomials are

$$\boldsymbol{A}(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{n_\mathrm{A}} q^{-n_\mathrm{A}}$$
$$\boldsymbol{B}(q^{-1}) = b_\tau q^{-\tau} + \ldots + b_{n_\mathrm{B}} q^{-n_\mathrm{B}}$$

The backward shift operator is denoted by q^{-1}, integers n_A, n_B, τ define the order of dynamics, $\tau \leq n_\mathrm{B}$. The output of the dynamic part is

$$x(k) = \sum_{l=\tau}^{n_\mathrm{B}} b_l u(k-l) - \sum_{l=1}^{n_\mathrm{A}} a_l x(k-l) \tag{4}$$

The nonlinear steady-state part of the model is described by the equation

$$y(k) = g(x(k))$$

where the function $g\colon \mathbb{R} \to \mathbb{R}$ is represented by the MultiLayer Perceptron (MLP) feedforward neural network with one hidden layer [2]. The network has one input, K nonlinear hidden nodes and one linear output. Its output is

$$y(k) = w_0^2 + \sum_{i=1}^{K} w_i^2 \varphi(w_{i,0}^1 + w_{i,1}^1 x(k)) \tag{5}$$

where $\varphi\colon \mathbb{R} \to \mathbb{R}$ is the nonlinear transfer function. Weights of the neural network are denoted by $w_{i,j}^1$, $i = 1, \ldots, K$, $j = 0, 1$ and w_i^2, $i = 0, \ldots, K$, for the first and the second layer, respectively.

The output of the neural Wiener model can be expressed as a function of input and auxiliary signal values at previous sampling instants

$$y(k) = f(u(k-\tau), \ldots, u(k-n_\mathrm{B}), x(k-1), \ldots, x(k-n_\mathrm{A})) \tag{6}$$

From (4) and (5) one has

$$y(k) = w_0^2 + \sum_{i=1}^{K} w_i^2 \varphi\left(w_{i,0}^1 + w_{i,1}^1 \left(\sum_{l=\tau}^{n_\mathrm{B}} b_l u(k-l) - \sum_{l=1}^{n_\mathrm{A}} a_l x(k-l)\right)\right) \tag{7}$$

4 MPC-NPLPT Algorithm with Neural Wiener Models

Typically, the majority of models used in MPC, for example neural models, are of Nonlinear Auto Regressive with eXternal input (NARX) type

$$y(k) = f(u(k-\tau), \ldots, u(k-n_{\mathrm{B}}), y(k-1), \ldots, y(k-n_{\mathrm{A}}))$$

A linear approximation of the NARX model can be easily calculated on-line for the current operating point

$$y(k) = \sum_{l=\tau}^{n_{\mathrm{B}}} b_l(k)u(k-l) - \sum_{l=1}^{n_{\mathrm{A}}} a_l(k)y(k-l) \tag{8}$$

where $a_l(k) = -\frac{\partial f(\cdot)}{\partial y(k-l)}$, $b_l(k) = \frac{\partial f(\cdot)}{\partial u(k-l)}$. The linearised model can be used for prediction in MPC, which leads to a quadratic programming MPC problem [6].

For the Wiener model (6) an inverse steady-state model

$$x(k) = g^{\mathrm{inv}}(y(k))$$

must be used to eliminate the auxiliary signal x. The Wiener model is first transformed into the NARX model

$$y(k) = f(u(k-1), \ldots, u(k-n_{\mathrm{B}}), g^{\mathrm{inv}}(y(k-1)), \ldots, g^{\mathrm{inv}}(y(k-n_{\mathrm{A}})))$$

Next, the obtained model can be linearised on-line and used for prediction in MPC [5]. Unfortunately, the class of processes for which the inverse model exists is limited. Typically, technological processes are characterised by saturations.

Alternatively, thanks to the cascade structure of the Wiener model, it can be linearised in an approximate way [4]. The time-varying gain $K(k)$ of the steady-state part of the model is estimated for the current operating point

$$y(k) = K(k)x(k)$$

The approximate linearisation for the current operating point can be expressed in the classical form (8), where time-varying coefficients of the model are

$$b_i(k) = K(k)b_i, \quad a_i(k) = a_i$$

and a_i, b_i are parameters of the linear part (3) of the model. Unfortunately, the model linearised in an approximate way may be significantly different from the original nonlinear Wiener model. In consequence, control accuracy of the resulting MPC algorithm may be insufficient.

4.1 Quadratic Programming MPC-NPLPT Optimisation Problem

The structure of the discussed MPC algorithm with Nonlinear Prediction and Linearisation along the Predicted Trajectory (MPC-NPLPT) is depicted in Fig. 2. In contrast to previously mentioned MPC algorithms, the neural model is not

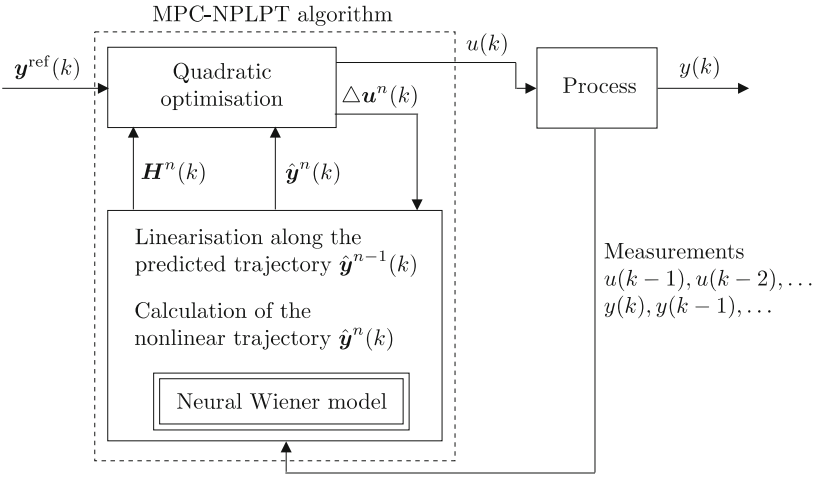


Fig. 2. The structure of the MPC algorithm with Nonlinear Prediction and Linearisation along the Predicted Trajectory (MPC-NPLPT) with the neural Wiener model

linearised once for the current operating point but for each sampling instant k linearisation is carried out in an iterative manner a few times.

In the n^{th} internal iteration the neural Wiener model is linearised along the trajectory $\boldsymbol{y}^{n-1}(k)$ calculated for the future control policy $\boldsymbol{u}^{n-1}(k)$ found in the previous internal iteration. As the initial future trajectory the control signal from the previous sampling instant, i.e. $\boldsymbol{u}^0(k) = [u(k-1) \ldots u(k-1)]^{\mathrm{T}}$, or the last $N_{\mathrm{u}} - 1$ elements of the optimal control policy calculated at the previous sampling instant can be used. Using the Taylor series expansion formula, one obtains a linear approximation of the predicted nonlinear trajectory $\hat{\boldsymbol{y}}^n(k)$

$$\hat{\boldsymbol{y}}^n(k) = \hat{\boldsymbol{y}}^{n-1}(k) + \boldsymbol{H}^n(k)(\boldsymbol{u}^n(k) - \boldsymbol{u}^{n-1}(k)) \tag{9}$$

where

$$\boldsymbol{H}^n(k) = \frac{\mathrm{d}\hat{\boldsymbol{y}}^{n-1}(k)}{\mathrm{d}\boldsymbol{u}^{n-1}(k)} = \begin{bmatrix} \dfrac{\partial \hat{y}^{n-1}(k+1|k)}{\partial u^{n-1}(k|k)} & \cdots & \dfrac{\partial \hat{y}^{n-1}(k+1|k)}{\partial u^{n-1}(k+N_{\mathrm{u}}-1|k)} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial \hat{y}^{n-1}(k+N|k)}{\partial u^{n-1}(k|k)} & \cdots & \dfrac{\partial \hat{y}^{n-1}(k+N|k)}{\partial u^{n-1}(k+N_{\mathrm{u}}-1|k)} \end{bmatrix}$$

is a matrix of dimensionality $N \times N_{\mathrm{u}}$, it consists of partial derivatives of the predicted output trajectory $\hat{\boldsymbol{y}}^{n-1}(k)$ with respect to the input trajectory $\boldsymbol{u}^{n-1}(k)$,

$$\boldsymbol{u}^{n-1}(k) = \left[u^{n-1}(k|k) \ldots u^{n-1}(k+N_{\mathrm{u}}-1|k)\right]^{\mathrm{T}}$$
$$\hat{\boldsymbol{y}}^{n-1}(k) = \left[\hat{y}^{n-1}(k+1|k) \ldots \hat{y}^{n-1}(k+N|k)\right]^{\mathrm{T}}$$

are vectors of length N_{u} and N, respectively, vectors $\boldsymbol{u}^n(k)$ and $\hat{\boldsymbol{y}}^n(k)$ are similar.

Since increments $\triangle \boldsymbol{u}^n(k)$ rather than $\boldsymbol{u}^n(k)$ are calculated, the relation

$$\boldsymbol{u}^n(k) = \boldsymbol{J}\triangle \boldsymbol{u}^n(k) + \boldsymbol{u}(k-1) \tag{10}$$

is used, $\boldsymbol{J}$ is a lower triangular matrix of dimensionality $N_\mathrm{u} \times N_\mathrm{u}$. Using (2), (9) and (10) the MPC-NPLPT quadratic programming task is formulated

$$\min_{\triangle \boldsymbol{u}^n(k)} \Big\{ \big\| \boldsymbol{y}^{\mathrm{ref}}(k) - \hat{\boldsymbol{y}}^{n-1}(k) - \boldsymbol{H}^n(k)\boldsymbol{J}\triangle \boldsymbol{u}^n(k) - \boldsymbol{H}^n(k)(\boldsymbol{u}(k-1) - \boldsymbol{u}^{n-1}(k)) \big\|^2 + \|\triangle \boldsymbol{u}^n(k)\|^2_{\boldsymbol{\Lambda}} \Big\}$$

subject to

$$\begin{aligned} &\boldsymbol{u}^{\min} \le \boldsymbol{J}\triangle \boldsymbol{u}^n(k) + \boldsymbol{u}(k-1) \le \boldsymbol{u}^{\max} \\ &-\triangle \boldsymbol{u}^{\max} \le \triangle \boldsymbol{u}^n(k) \le \triangle \boldsymbol{u}^{\max} \\ &\boldsymbol{y}^{\min} \le \hat{\boldsymbol{y}}^{n-1}(k) + \boldsymbol{H}^n(k)\boldsymbol{J}\triangle \boldsymbol{u}^n(k) + \boldsymbol{H}^n(k)(\boldsymbol{u}(k-1) - \boldsymbol{u}^{n-1}(k)) \le \boldsymbol{y}^{\max} \end{aligned}$$

In the quadratic programming problem $\boldsymbol{y}^{\mathrm{ref}} = \left[y^{\mathrm{ref}}(k+1|k) \ldots y^{\mathrm{ref}}(k+N|k)\right]^{\mathrm{T}}$, $\boldsymbol{y}^{\min} = \left[y^{\min} \ldots y^{\min}\right]^{\mathrm{T}}$ and $\boldsymbol{y}^{\max} = \left[y^{\max} \ldots y^{\max}\right]^{\mathrm{T}}$ are vectors of length N, $\boldsymbol{u}^{\min} = \left[u^{\min} \ldots u^{\min}\right]^{\mathrm{T}}$, $\boldsymbol{u}^{\max} = \left[u^{\max} \ldots u^{\max}\right]^{\mathrm{T}}$, $\triangle \boldsymbol{u}^{\max} = \left[\triangle u^{\max} \ldots \triangle u^{\max}\right]^{\mathrm{T}}$ and $\boldsymbol{u}(k-1) = \left[u(k-1) \ldots u(k-1)\right]^{\mathrm{T}}$ are vectors of length N_u, the diagonal matrix $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda, \ldots, \lambda)$ is of dimensionality $N_\mathrm{u} \times N_\mathrm{u}$.

If the the operating point does not change significantly, it would be sufficient to carry out only one internal iteration. Internal iterations are continued if

$$\sum_{p=0}^{N_0} (y^{\mathrm{ref}}(k-p) - y(k-p))^2 \ge \delta_\mathrm{y}$$

If $\left\|\triangle \boldsymbol{u}^n(k) - \triangle \boldsymbol{u}^{n-1}(k)\right\|^2 < \delta_\mathrm{u}$ or $n > n_{\max}$ internal iterations are terminated. Quantities δ_u, δ_y, N_0 and $n_{\max}$ are adjusted by the user.

4.2 Implementation Details

The nonlinear output trajectory $\hat{y}^n(k+p|k)$ is calculated on-line recurrently over the prediction horizon (for $p = 1, \ldots, N$) from the neural Wiener model (5)

$$\hat{y}^n(k+p|k) = w_0^2 + \sum_{i=1}^{K} w_i^2 \varphi(w_{i,0}^1 + w_{i,1}^1 x^n(k+p|k)) + d(k) \tag{11}$$

where from (4)

$$\begin{aligned} x^n(k+p|k) = &\sum_{j=1}^{I_{\mathrm{uf}}(p)} b_j u^n(k-\tau+1-j+p|k) + \sum_{j=I_{\mathrm{uf}}(p)+1}^{I_\mathrm{u}} b_j u(k-\tau+1-j+p) \\ &- \sum_{j=1}^{I_{\mathrm{yp}}(p)} a_j x^n(k-j+p|k) - \sum_{j=I_{\mathrm{yp}}(p)+1}^{n_\mathrm{A}} a_j x(k-j+p) \end{aligned} \tag{12}$$

and $I_{\mathrm{uf}}(p) = \max(\min(p-\tau+1, I_{\mathrm{u}}), 0)$, $I_{\mathrm{u}} = n_{\mathrm{B}} - \tau + 1$, $I_{\mathrm{yp}}(p) = \min(p-1, n_{\mathrm{A}})$. The unmeasured disturbance is estimated as the difference between the measured process output and the output value calculated from the model (7)

$$d(k) = y(k) - w_0^2 - \sum_{i=1}^{K} w_i^2 \varphi\left(w_{i,0}^1 + w_{i,1}^1 \left(\sum_{l=\tau}^{n_{\mathrm{B}}} b_l u(k-l) - \sum_{l=1}^{n_{\mathrm{A}}} a_l x(k-l)\right)\right)$$

Using (11) and (12) (where n must be replaced by $n-1$), entries of the matrix $\boldsymbol{H}^n(k)$ are calculated for $p = 1, \ldots, N$, $r = 0, \ldots, N_{\mathrm{u}} - 1$ from

$$\frac{\partial \hat{y}^{n-1}(k+p|k)}{\partial u^{n-1}(k+r|k)} = \sum_{i=1}^{K} w_i^2 \frac{\partial \varphi(z_i^{n-1}(k+p|k))}{\partial z_i^{n-1}(k+p|k)} w_{i,1}^1 \frac{\partial x^{n-1}(k+p|k)}{\partial u^{n-1}(k+r|k)}$$

where $z_i^{n-1}(k+p|k)) = w_{i,0}^1 + w_{i,1}^1 x^{n-1}(k+p|k)$. If hyperbolic tangent is used as the function φ, $\frac{\partial \varphi(z_i^{n-1}(k+p|k))}{\partial z_i^{n-1}(k+p|k)} = 1 - \tanh^2(z_i^{n-1}(k+p|k))$. Derivatives are

$$\frac{\partial x^{n-1}(k+p|k)}{\partial u^{n-1}(k+r|k)} = \sum_{j=1}^{I_{\mathrm{uf}}(p)} b_j \frac{\partial u^{n-1}(k-\tau+1-j+p|k)}{\partial u^{n-1}(k+r|k)} - \sum_{j=1}^{I_{\mathrm{yp}}(p)} a_j \frac{\partial x^{n-1}(k-j+p|k)}{\partial u^{n-1}(k+r|k)}$$

5 Simulation Results

The linear part of the Wiener system under consideration is described by

$$\boldsymbol{A}(q^{-1}) = 1 - 0.9744q^{-1} + 0.2231q^{-2}, \quad \boldsymbol{B}(q^{-1}) = 0.3096q^{-1} + 0.1878q^{-2} \quad (13)$$

The nonlinear steady-state part of the system is shown in Fig. 3. It represents a valve with saturation, its inverse function does not exist. The Wiener system described by (13) and the characteristics shown in Fig. 3 is a simulated process.

The following MPC algorithms are compared:

a) the classical MPC algorithm based on the linear model,
b) the MPC-NPAL algorithm (with approximate linearisation) based on the neural Wiener model and quadratic programming [4],
c) the described MPC-NPLPT algorithm based on the neural Wiener model and quadratic programming,
d) the MPC-NO algorithm with on-line nonlinear optimisation, it uses the same neural Wiener model. It is the "ideal" control algorithm.

In nonlinear MPC algorithms the same neural approximation with $K = 5$ hidden nodes (Fig. 3) of the steady-state part is used. Parameters of all algorithms are the same: $N = 10$, $N_{\mathrm{u}} = 2$, $\lambda = 0.2$, $u^{\min} = -5$, $u^{\max} = 5$, $\triangle u^{\max} = 2.5$, for the MPC-NPLPT algorithm: $\delta_{\mathrm{u}} = \delta_{\mathrm{y}} = 10^{-1}$, $N_0 = 2$, $n_{\max} = 5$.

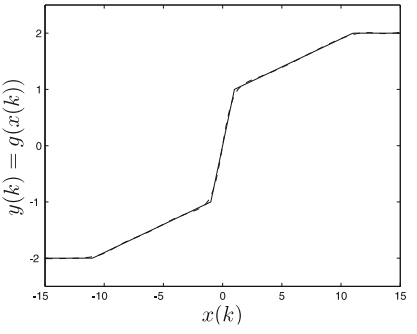


Fig. 3. The characteristics $y(k) = g(x(k))$ of the steady-state part of the process (*solid line*) and its neural approximation (*dashed line*)

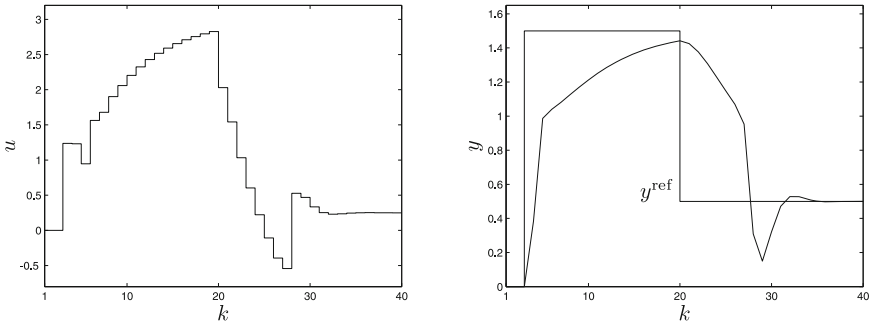


Fig. 4. Simulation results of the MPC algorithm based on the linear model

Because the process is significantly nonlinear, the linear MPC algorithm does not work properly as depicted in Fig. 4. The MPC-NPAL algorithm with approximate linearisation is significantly faster and gives much smaller overshoot as shown in Fig. 5. On the other hand, unfortunately, closed-loop performance of the MPC-NPAL algorithm is reasonably different from that of the MPC-NO approach. In particular, when the reference trajectory changes from 1.5 to 0.5 the algorithm with approximate linearisation gives unwanted oscillations.

Fig. 6 compares MPC-NO and MPC-NPLPT algorithms. Unlike the MPC-NPAL algorithm, the discussed MPC-NPLPT strategy gives trajectories practically the same as the "ideal" MPC-NO approach (very small differences are visible for sampling instants $k = 30, \ldots, 33$). At the same time, the MPC-NPLPT algorithm is approximately 4.5 times more computationally efficient than the MPC-NO approach. Fig. 7 shows the number of internal iterations in consecutive main iterations (sampling instants) of the MPC-NPLPT algorithm. When the current value of the output is significantly different from the reference, the algorithm needs two or three internal iterations, but for some 50% of the simulation one iteration is sufficient.

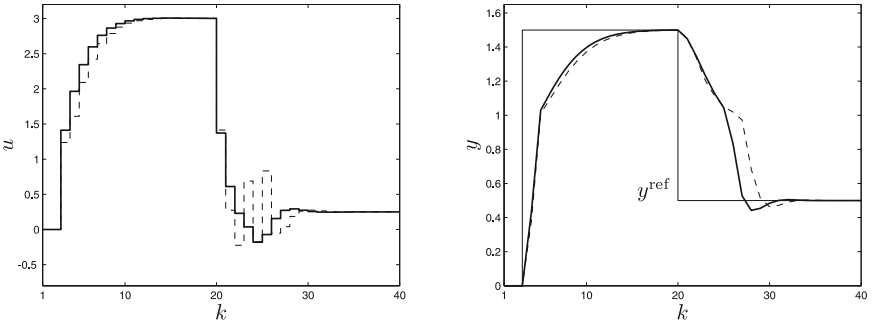


Fig. 5. Simulation results: the MPC-NO algorithm with nonlinear optimisation based on the neural Wiener model (*solid line*) and the MPC-NPAL algorithm with approximate linearisation and quadratic programming based on the same model (*dashed line*)

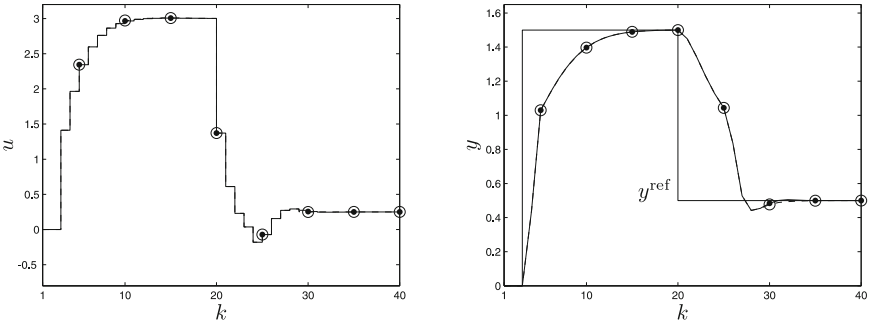


Fig. 6. Simulation results: the MPC-NO algorithm with nonlinear optimisation based on the neural Wiener model (*solid line with dots*) and the MPC-NPLPT algorithm with quadratic programming based on the same model (*dashed line with circles*)

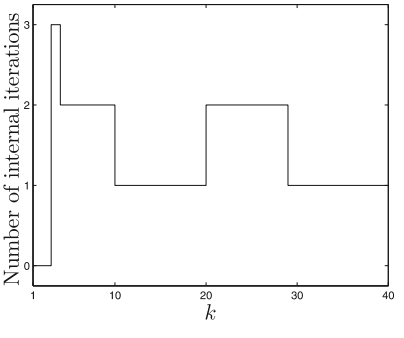


Fig. 7. The number of internal iterations in consecutive main iterations of the MPC-NPLPT algorithm

6 Conclusions

The described MPC-NPLPT algorithm uses for prediction a linear approximation of the neural Wiener model which is carried out along the predicted trajectory. For on-line optimisation quadratic programming is used. Unlike existing MPC algorithms based on the Wiener model, the inverse of the steady-state part of the model is not used. Hence, it can be used when the inverse function does not exist. The algorithm gives better control than the MPC scheme with approximate linearisation. Moreover, its control accuracy is practically the same as that of the computationally expensive algorithm with on-line nonlinear optimisation.

Acknowledgement. The work presented in this paper was supported by Polish national budget funds for science.

References

1. Cervantes, A.L., Agamennoni, O.E., Figueroa, J.L.: A nonlinear model predictive control based on Wiener piecewise linear models. Journal of Process Control 13, 655–666 (2003)
2. Haykin, S.: Neural networks–a comprehensive foundation. Prentice Hall, Englewood Cliffs (1999)
3. Janczak, A.: Identification of nonlinear systems using neural networks and polynomial models: block oriented approach. Springer, London (2004)
4. Ławryńczuk, M.: Nonlinear predictive control based on multivariable neural Wiener models. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part I. LNCS, vol. 6593, pp. 31–40. Springer, Heidelberg (2011)
5. Ławryńczuk, M.: Computationally efficient nonlinear predictive control based on neural Wiener models. Neurocomputing 74, 401–417 (2010)
6. Ławryńczuk, M.: A family of model predictive control algorithms with artificial neural networks. International Journal of Applied Mathematics and Computer Science 17, 217–232 (2007)
7. Maciejowski, J.M.: Predictive control with constraints. Prentice Hall, Harlow (2002)
8. Mahfouf, M., Linkens, D.A.: Non-linear generalized predictive control (NLGPC) applied to muscle relaxant anaesthesia. International Journal of Control 71, 239–257 (1998)
9. Morari, M., Lee, J.H.: Model predictive control: past, present and future. Computers and Chemical Engineering 23, 667–682 (1999)
10. Nørgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural networks for modelling and control of dynamic systems. Springer, London (2000)
11. Norquay, S.J., Palazoğlu, A., Romagnoli, J.A.: Model predictive control based on Wiener models. Chemical Engineering Science 53, 75–84 (1998)
12. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Engineering Practice 11, 733–764 (2003)
13. Tatjewski, P.: Advanced control of industrial processes, Structures and algorithms. Springer, London (2007)

Adaptive Immunization in Dynamic Networks

Jiming Liu and Chao Gao

Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, HK
jiming@comp.hkbu.edu.hk

Abstract. In recent years, immunization strategies have been developed for stopping epidemics in complex-network-like environments. So far, there exist two limitations in the current propagation models and immunization strategies: (1) the propagation models focus only on the network structure underlying virus propagation and the models are static; (2) the immunization strategies are offline and non-adaptive in nature, i.e., these strategies pre-select and pre-immunize "important" nodes before virus propagation starts. In this paper, we extend an interactive email propagation model in order to observe the effects of human behaviors on virus propagation, and furthermore we propose an adaptive AOC-based immunization strategy for protecting dynamically-evolving email networks. Our experimental results have shown that our strategy as an online strategy can adapt to the dynamic changes (e.g., growth) of networks.

1 Introduction

Currently, there have been many studies on modeling virus propagation, including agent-based models [1][2] and population-based models [3][4], and on designing effective immunization strategies for restraining virus propagation [5][6][7][8][9][10]. These propagation models have provided feasible test-beds for examining the mechanisms of virus propagation and for evaluating new and/or improved security strategies for restraining virus propagation [2]. However, there exist some limitations in the current work. For example, the proposed models of epidemic dynamics and immunization strategies are static in nature, i.e., the connections of networks are treated as being constant over time, and the immunization strategies are pre-immunization-based and non-adaptive to dynamically evolving networks.

The assumption about static networks may be reasonable in some cases of network modeling, as changes in the connections (edges) among individuals (nodes) may evolve slowly with respect to the speed of virus propagation. However, in a longer term, the changes of network structures should be taken into consideration. As pointed out by Keeling et al., the behaviors of virus population may dramatically change as a result of infection outbreaks [11], which needs to be considered when designing intervention strategies. Furthermore, with the growing applications of mobile phones and GPS, it has become necessary for us to accurately track the movements of people in real time and to model the changing nature of network structures in the face of a severe epidemic.

Several recently developed immunization strategies, e.g., acquaintance [5][6], targeted [7], D-steps [8], AOC-based strategies [9][10], have shared one commonality in that they first select certain nodes to immunize that have high degrees of connectivity in

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 673–683, 2011.

a network, and then simulate viruses spreading with a typical propagation model. Here, we refer to this type of strategies as static and offline pre-immunization strategies. Fig. 3 in [2] provides an illustration of the process of a static strategy. These strategies will be able to protect some important nodes in advance, by cutting epidemic paths. However, in some cases, we can observe and detect certain viruses only after these viruses have already propagated in a network (e.g., "Melissa", "W32/Sircam"). Therefore, it would be desirable for us to develop an online and adaptive strategy in order to fast dispatch antivirus programs or vaccines into a network and hence restrain virus propagation, or even kill these viruses and recover the network.

In this paper, we extend a distributed network immunization strategy based on Autonomy-Oriented Computing (AOC), as reported in [9][10]. The extended AOC-based strategy can adapt to dynamically-evolving networks and restrain virus propagation as an online strategy. Then, we use an improved interactive email propagation model [2], as a test-bed to evaluate whether the adaptive AOC-based strategy can protect dynamically-evolving email networks.

The remainder of this paper is organized as follows: Section 2 states our research questions. Section 3 presents the basic ideas and the detailed formulations of our proposed strategy. Section 4 provides several experiments for systematically validating our strategy. Section 5 highlights our major contributions and future work.

2 Problem Statements

In this paper, we focus on the network immunization strategies for restraining virus propagation in email networks. This section provides general definitions about an email network and a network immunization strategy.

Definition 1. *A graph $G = \langle V, L\rangle$ is an **email network** based on address books, where $V=\{v_1, v_2, ..., v_N\}$ is a set of nodes, and $L = \{\langle v_i, v_j\rangle \mid 1 \le i, j \le N, i \ne j\}$ is a set of undirected links (if v_i in the address book of v_j, there exists a link between v_i and v_j). $N = |V|$ and $E = |L|$ represent the total numbers of nodes and edges in the email network, respectively.*

Each user in an email network has two operational behaviors: checking an email and opening an email. By analyzing the email-checking intervals in the Enron email dataset [2] and related studies on human dynamics [12][13][14], we have found that the email-box checking intervals of a user follow a power-law distribution with a long tail. Based on these findings, we improve an interactive email propagation model. Different from the traditional interactive email model [1], our interactive email model incorporates two further changes: (1) The email-checking intervals of a user follow a power-law distribution based on our previous research on human dynamics [2]; (2) We extend the states of each node in the interactive email model [1][2]. Each node has five states: (1) Healthy, (2) Danger, (3) Infected, (4) Immunized and (5) Recoverable. In our model, the states of each node will be updated based on the following rules:

- Healthy→Danger: The state of a node changes from Healthy to Danger, if a user receives an email with a virus-embedded email attachment;

- Danger→Infected: The state of a node changes from Danger to Infected, if a user checks its email-box and clicks on a virus-embedded email attachment;
- Danger→Healthy: The state of a node changes from Danger to Healthy, if a user checks its email-box, but does not click on a virus-embedded email attachment;
- Infected→Recoverable: If an entity with vaccines reaches to Infected nodes, it will kill viruses and help recover the nodes;
- Healthy, Danger→Immunized: If an entity with vaccines reaches to Healthy or Danger nodes, it will inject vaccines into the nodes and protect them from attacks.

Network immunization [9][10] is one of the most popular methods for restraining virus propagation and providing network security. For a network, some immunization strategies select and protect certain *important* nodes from being infected by cutting epidemic paths. For such a network, the vaccinated nodes are denoted as $V_e \subseteq V$, where $|V_e| = N_e$ and $N_e \leq N$.

Definition 2. *A **pre-immunization strategy** selects and protects a set of nodes V_e, denoted as $V_e = P\_IS(V_0, G)$, where $V_0 \subseteq V$ is an initial set of "seed" nodes, which correspond to the initial positions of entities in our strategy. The output V_e indicates the final positions of the entities, i.e., a set of* important *nodes to be immunized.*

In network immunization, a common criterion used to evaluate the efficiency of different strategies is to measure the total numbers of infected nodes after virus propagation, i.e., $N_{Infected} = |V_{Infected}|$. However, the current strategies are non-adaptive and offline in nature; they pre-select immunized nodes before virus propagation in some static networks and just protect those *important* nodes from being infected rather than recover infected nodes. In this paper, we propose an adaptive and online strategy by extending the AOC-based immunization strategy, as reported in [9][10], in order to recover infected nodes even in a dynamically-evolving network.

Definition 3. *An **adaptive AOC-based immunization strategy** refers to a scheme for forwarding vaccines from initial "seed" nodes (V_0) to more unprotected nodes ($V_{patched}$) in a network. Autonomous entities with vaccines travel in a network based on their own local behaviors. These entities will recover those infected nodes and/or immunize those susceptible nodes that they meet.*

In this paper, we utilize an improved interactive email model [2] as a test-bed to evaluate the efficiency of the adaptive AOC-based immunization strategy in both static benchmark and dynamically-evolving synthetic networks. Specifically, we conduct some experiments to observe the effects of dynamic networks on virus propagation and the corresponding performance of our immunization strategy. Specific research questions to be answered are as follows:

1. What are the process and characteristics of virus propagation in dynamically-evolving networks? What are the effects of human dynamics in virus propagation?
2. Can the adaptive AOC-based immunization strategy fast restrain virus propagation in different types of networks? In other words, can the adaptive AOC-based immunization strategy efficiently forward vaccines into a network in order to immunize susceptible nodes and/or recover infected nodes?

3 An Adaptive AOC-Based Immunization Strategy

In the real world, antivirus programs, system patches, and vaccines are dispatched into networks after certain viruses have propagated and detected. In this section, we present an adaptive AOC-based immunization strategy for dynamically disseminating security information (i.e., defense techniques) to the nodes of a network.

Previous studies have proven and illustrated that vaccinating *high-degree* nodes in a network can effectively restrain virus propagation [10][11]. Based on the ideas of positive-feedback and self-organization, we have introduced a distributed immunization strategy based on Autonomy-Oriented Computing (AOC) in [9][10]. The AOC-based strategy has shown to be capable of restraining virus propagation by means of immunizing a set of highly-connected nodes (i.e., the highest-degree nodes) in a network. In our current work, we further extend the AOC-based immunization strategy in order to dynamically and adaptively immunize and/or recover the whole network. Since the page limitation, the methodology of Autonomy-Oriented Computing (AOC) for network immunization are fully introduced and discussed in [9][10].

It should be noted that the aim of our previous AOC-based strategy is to find a set of highly-connected nodes in a network (i.e., a distributed search problem). However, the task to be accomplished by new adaptive strategy is to disseminate vaccines (security information or patches) to as many nodes as possible (i.e., a route selection problem).

In our proposed adaptive AOC-based immunization strategy, autonomous entities with vaccines will travel in a network in order to efficiently disseminate security information or patches to other nodes. Based on our previous research, these entities will first move to high-degree nodes in order to visit and patch more nodes with a lower transferring cost. If the encountered nodes have already been infected by certain viruses, the entities will help recover them from the infected state (Infected→Recoverable). If the nodes are susceptible but have not been infected by any viruses, the entities will inject them with vaccines in order to protect them from certain viruses (Healthy→Immunized).

In the adaptive AOC-based immunization strategy, each autonomous entity will have three Behaviors: Rational-move, Random-jump, and Wait.

1. Rational-move: Our previous work has proved that a set of autonomous entities can find a set of the highest degree nodes in a few steps based on their self-organization computing and the positive feedback mechanism [9][10]. One of characteristics of previous strategy is that an entity will stay at the highest-degree node in its local environment and not move any more. In our new adaptive AOC-based strategy, however, even if an entity has found and resided in the highest-degree node in its local environment, it will continue to move the highest-degree neighbor, which has not been resided before. If there exist more than one highest-degree positions in its neighborhood, the entity will choose the first one from its friend list.
2. Random-jump: An entity moves along the connected edges of a network, with a randomly-determined number of hops (steps), in order to escape a local optimum.
3. Wait: If an entity does not find any nodes available for the entity to reside in, the entity will stay at the old place.

The behavioral rules for activating the above behaviors are given in Algorithm 1. Definitions about the above-mentioned autonomous entities and their global and local

Algorithm 1. The behavioral rules of autonomous entities

Input: e_i and its local information
Output: the behavior of e_i
1. **For** each entity e_i
2. compute $targetId$ based on $e_i.E_l$;
3. **If** $targetId$ is not *null* **then**
4. $e_i.Rational\_Move(targetId)$;
5. **Else if** $e_i.lifecycle < 1$ **then**
6. $e_i.Random\_jump(targetId)$;
7. **Else**
8. $e_i.Wait()$;

environments (e.g., $e_i.E_l$ and a friend list) are the same as those for the previous AOC-based strategy. The algorithm complexity is also close to our previous strategy. Due to the space limitation, we will not describe them here; details can be found in [9][10].

Figure 1 presents an illustrative example of the adaptive AOC-based immunization strategy. Suppose that in a network, two entities (i.e., e_1 and e_2) have been randomly deployed (i.e., at v_8 and v_4). Each entity has its own local environment, which is composed of direct and indirect neighbors [9][10]. For example, e_1 only has direct neighbors (i.e., v_1, v_2, v_5, v_7, v_8, v_9, and v_{11}), whereas e_2 has direct neighbors (i.e., v_1, v_3, v_4, and v_6) and an indirect neighbor (i.e., v_9), as shown in Fig. 1(a). In the next step, e_1 will move to the non-resided highest-degree node (i.e., v_{11}), even if e_1 has already resided in the highest-degree node within its local environment, which is different from the previous strategy. More importantly, the core of the adaptive AOC-based strategy lies in its positive-feedback mechanism. When e_1 moves from v_8 to v_6 as shown in Fig. 1(a), there is an interaction between e_1 and e_2. Through this (sharing) interaction, e_2 enlarges its local environment, and further finds an indirect neighbor, i.e., v_9. In the next step, e_2 moves from v_2 to v_9. Based on this coupling relationship, an entity can dispatch vaccines to more nodes, and help recover them.

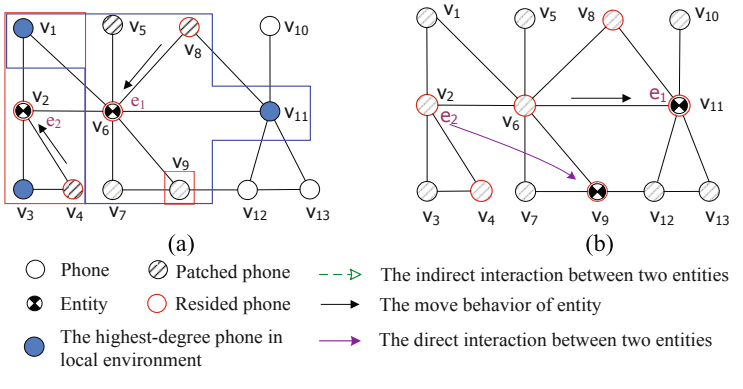


Fig. 1. An illustrative example of the adaptive AOC-based immunization strategy

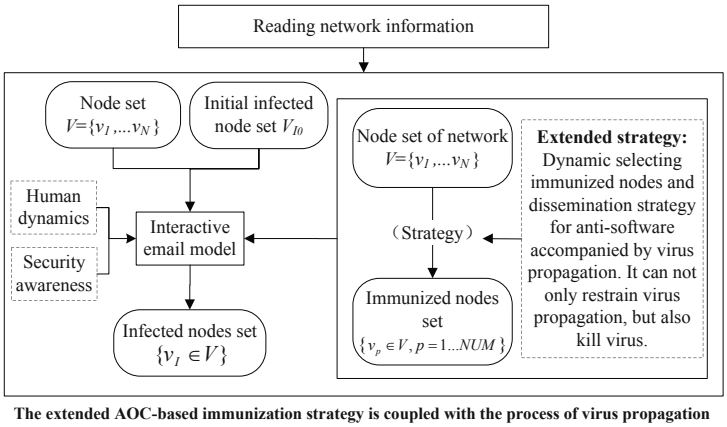


Fig. 2. The process of the adaptive AOC-based immunization

4 Experiments

In this section, we first examine the effects of human behaviors on virus propagation in the improved interactive email model. Then, we present several experiments for evaluating the efficiency of the adaptive AOC-based strategy in both benchmark and synthetic growing networks. Different from the previous static pre-immunization strategy [9][10], the adaptive AOC-based strategy deploys vaccines into a network after viruses have been spread, as shown in Fig. 2. That is to say, the adaptive AOC-based strategy is an online strategy, which will adapt to the dynamic changes of a network.

The following are some assumptions about the interactive email model:

1. If a user opens an infected email, the corresponding node will be infected and thus it will send viruses to all its friends based on its hit-list;
2. When checking his/her mailbox, if a user does not click on virus-embedded emails, we assume that the user will delete those suspected emails;
3. If nodes are immunized, they will never send viruses even if a user clicks on virus-embedded email attachments.

At the beginning, we randomly select two nodes as the initially-infected nodes in a network (i.e., based on a random attack model). The adaptive AOC-based immunization strategy is triggered at step=50. All experimental results are given in average values having simulated for 100 times.

4.1 The Structures of Experimental Email Networks

We use four benchmark networks to evaluate our adaptive strategy; they are: coauthorship network (NET1)[1], autonomous system network (NET2), an university email network (NET3)[2], and the Enron email network (NET4).

[1] http://www-personal.umich.edu/∼mejn/netdata/astro-ph.zip
[2] http://deim.urv.cat/∼aarenas/data/welcome.htm

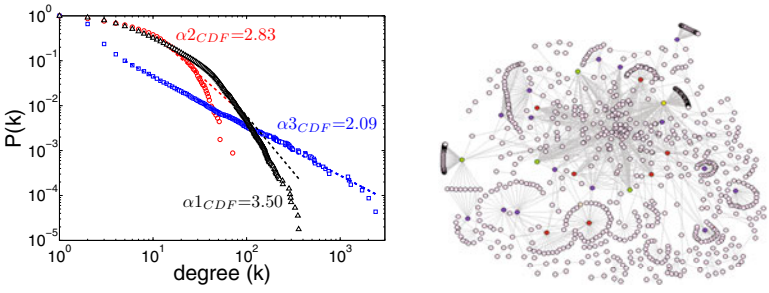


Fig. 3. (a) The cumulative degree distributions of three benchmark networks. α is the maximum likelihood power-law exponent based on [17]. (b) The structure of the Enron email network.

NET1 was built based on the published papers on the widely-used Physics E-print Archive at arxiv.org. Meanwhile, Newman has pointed out that the coauthorship network consists of about 600 small communities and 4 large communities [15]. If people collaborate with each other to write a paper, they are very likely to know each other's email addresses. Thus, the structure of the coauthorship network can, to a certain extent, reflect social interaction. NET2 was generated from the University of Oregon Route Views Project[3] and the snapshot was created by Newman in 2006[4]. NET3 was complied by the members of University Rovira i Virgili (Tarragona) [16]. NET4 was released by Andrew Fiore and Jeff Heer[5]. Fig. 3(a) shows the cumulative degree distributions of NET1, NET2, and NET3, respectively. And, Fig. 3(b) presents the structure of NET4.

In addition, we have constructed some dynamically-evolving networks based on the GLP algorithm [18], in order to evaluate the efficiency of our strategy. Section 4.3 presents more results about the dynamic networks with various growing trends.

4.2 The Static Email Networks

First, we utilize four benchmark networks to evaluate the performance of the adaptive AOC-based immunization strategy. Then, we observe the effects of email-checking intervals on virus propagation in the coauthorship network.

We randomly deploy a few autonomous entities into a network at step=50 after viruses have propagated. As shown in Fig. 4, virus exhibits two spreading phases: (1) viruses have an explosive growth within 50 steps, since there is no immunization strategy deployed in a network; (2) viruses will decline after we deploy the adaptive AOC-based strategy at step=50. The simulation results confirm that the adaptive AOC-based strategy can effectively restrain virus propagation and recover the whole network.

In order to further observe the effects of human behaviors on virus propagation. We have used two distributions to depict a user's email-checking intervals. That is to say, the

[3] http://routeviews.org/

[4] http://www-personal.umich.edu/∼mejn/netdata/as-22july06.zip

[5] http://bailando.sims.berkeley.edu/enron/enron.sql.gz

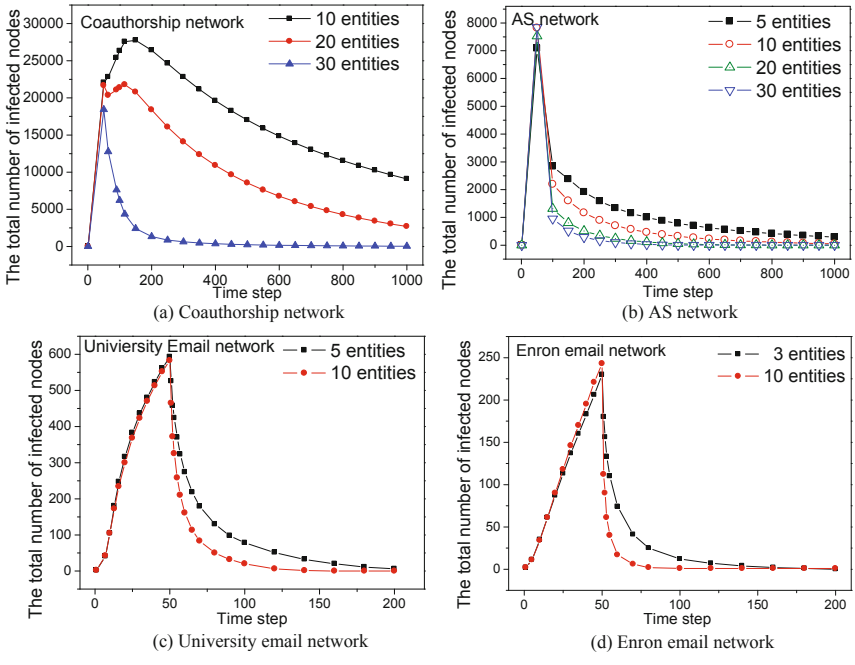


Fig. 4. The effects of the adaptive AOC-based strategy on virus propagation in static networks

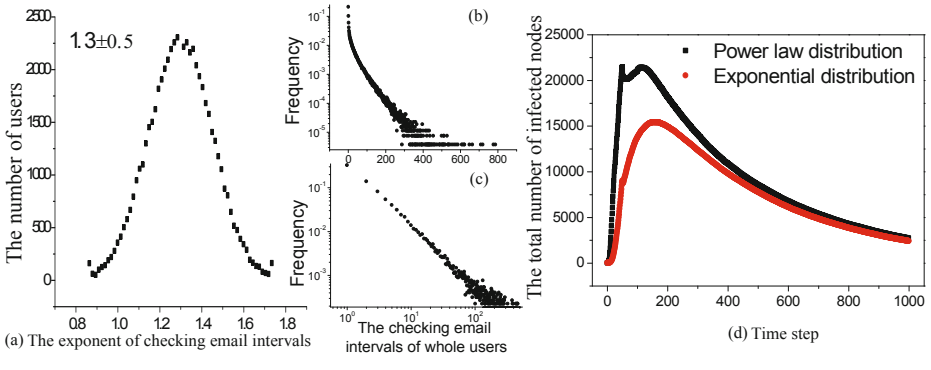


Fig. 5. (a) The power-law exponents of different users follow a normal distribution. (b)(c) Users' email-checking intervals follow an exponential distribution and a power-law distribution, respectively. (b) is a log-linear figure and (c) is a log-log figure. (d) The effects of users' email-checking intervals on virus propagation in the coauthorship network, with 20 entities.

email-checking intervals of a user follow either an exponential distribution or a power-law distribution with a long tail. But, the power-law exponents of different users follow a Gaussian distribution if the numbers of users are very large and users' behaviors are independent of each other [1]. Fig. 5(a) shows the power-law exponent distribution of different users. The distribution exponent (i.e., $\alpha \approx 1.3 \pm 0.5$) is based on our previous research [2]. Fig. 5(b)(c) provide two distributions of users' email-checking intervals that we will examine.

Figure 5(d) presents the effects of users' email-checking intervals on virus propagation. The simulation results reveal that viruses can fast spread in a network, if users' email-checking intervals follow a power-law distribution. In such a situation, viruses can have an explosive (acute) growth at the initial stage, and then a slower growth. That is because viruses will stay at a latent state and await activation by users [2].

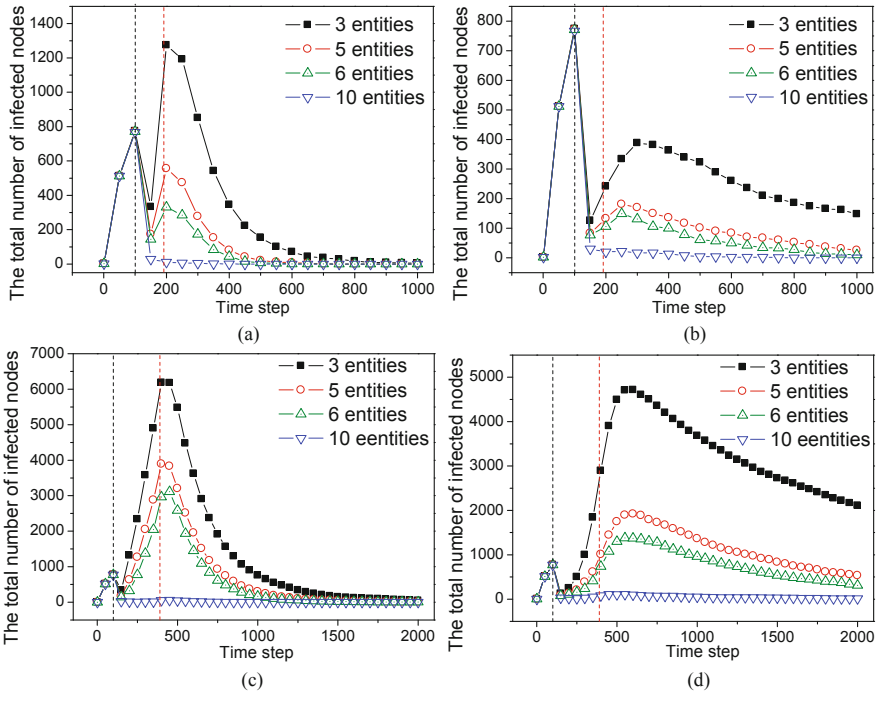


Fig. 6. The effects of the adaptive AOC-based strategy on virus propagation in dynamically-evolving networks. 100 nodes are added into a network at each step. (a)(b) The network scale increases from 10^3 to 10^4. (c)(d) The network scale increases from 10^3 to $3 * 10^4$. The average degree <K> increases from 8 to 33 in (a)(c), and maintains at 8 in (b)(d), respectively.

4.3 The Dynamic Email Networks

In the real world, the structure of a network can dynamically change all the time. In this regard, we have generated some synthetic growing networks based on the GLP

algorithm [18], in order to experimentally evaluate whether or not the adaptive AOC-based strategy can restrain virus propagation and recover the whole network.

The synthetic networks to be used in our experiments have various growing trends. They are: (1) the network scale increases from 10^3 to 10^4, or to $3 * 10^4$; (2) the average degree of a network increases from 8 to 33, or maintains at 8.

We randomly deploy different numbers of entities into a dynamically-evolving network at step=50. Fig. 6 shows that the adaptive AOC-based immunization strategy can effectively protect a network from the potential damages of email viruses even when the network dynamically evolves as mentioned. As can be noted from Fig. 6, there are two peak values in the case of growing networks. The first peak value is at step=50, which means that the adaptive AOC-based strategy starts to restrain virus propagation. Although viruses decline in the following time steps, viruses will outbreak again as the network grows. Fortunately, the adaptive AOC-based strategy can suppress the second peak, if we deploy an enough number of entities.

5 Conclusion

This paper has presented an online and adaptive strategy for restraining virus propagation in both benchmark and synthetic growing networks based on our previous work [9][10]. With the extended AOC-based strategy, entities can dispatch vaccines to most of nodes in a network in order to protect susceptible nodes from being infected and recover infected nodes. We have evaluated our proposed strategy on the improved interactive email model [2] in this paper, and on a mobile model [19] (The results are not reported here for the page limitation). The simulation-based experimental results all have shown that our new strategy can effectively protect both static and dynamic networks.

References

1. Zou, C.C., Towsley, D., Gong, W.: Modeling and simulation study of the propagation and defense of internet e-mail worms. IEEE Transaction on Dependable and Secure Computing 4(2), 105–118 (2007)
2. Gao, C., Liu, J., Zhong, N.: Network immunization and virus propagation in email networks: experimental evaluation and analysis. Knowledge and Information Systems 27(2), 253–279 (2011)
3. Lloyd, A.L., May, R.M.: How viruses spread among computers and people. Science 292(5520), 1316–1317 (2001)
4. Liu, J., Xia, S.: Effective epidemic control via strategic vaccine deployment: A systematic approach. In: Proceedings of the 1st ACM International Health Informatics Symposium (IHI 2010), pp. 91–99 (2010)
5. Cohen, R., Havlin, S., Ben-Averaham, D.: Efficient immunization strategies for computer networks and populations. Physical Review Letters 91(24), 247901 (2003)
6. Gallos, L.K., Liljeros, F., Argyrakis, P., Bunde, A., Havlin, S.: Improving immunization strategies. Physical Review E 75(4), 045104 (2007)
7. Chen, Y., Paul, G., Havlin, S., Liljeros, F., Stanley, H.E.: Finding a better immunization strategy. Physical Review Letters 101(5), 058701 (2008)

8. Echenique, P., Gomez-Gardenes, J., Moreno, Y., Vazquez, A.: Distance-d covering problem in scale-free networks with degree correlation. Physical Review E 71(3), 035102 (2005)
9. Liu, J., Gao, C., Zhong, N.: A distributed immunization strategy based on autonomy-oriented computing. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 503–512. Springer, Heidelberg (2009)
10. Gao, C., Liu, J., Zhong, N.: Network immunization with distributed autonomy-oriented entities. IEEE Transactions on Parallel and Distributed Systems (2010), doi: 10.1109/TPDS.2010.197
11. Keeling, M.J., Eames, K.T.: Networks and epidemic models. Journal of the Royal Society Interface 2(4), 295–307 (2005)
12. Barabasi, A.L.: The origin of bursts and heavy tails in human dynamics. Nature 435(7039), 207–211 (2005)
13. Eckmann, J.P., Moses, E., Sergi, D.: Entropy of dialogues creates coherent structure in email traffic. Proceedings of the National Academy of Sciences of the United States of America 101(40), 14333–14337 (2004)
14. Malmgren, R.D., Stouffer, D.B., Campanharo, A.S., Amaral, L.A.N.: On universality in human correspondence activity. Science 325(5948), 1696–1700 (2009)
15. Newman, M.E.J.: The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences of the United States of America 98(2), 404–409 (2001)
16. Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., Arenas, A.: Self-similar community structure in a network of human interactions. Physical Review E 68(6), 065103 (2003)
17. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distribution in empirical data. SIAM Review 51(4), 661–703 (2009)
18. Bu, T., Towsley, D.: On distinguishing between internet power law topology generators. In: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), pp. 638–647 (2002)
19. Gao, C., Liu, J.: Modeling and predicting the dynamics of mobile virus spread affected by human behavior. In: Proceedings of the 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2011 (in press, 2011)

A Memetic Algorithm for a Tour Planning in the Selective Travelling Salesman Problem on a Road Network

Anna Piwońska and Jolanta Koszelew

Department of Computer Science
Technical University of Bialystok
Wiejska 45A, 15-351 Bialystok, Poland
{a.piwonska,j.koszelew}@pb.edu.pl

Abstract. The selective travelling salesman problem (STSP) appears in various applications. The paper presents a new version of this problem called the selective travelling salesman problem on a road network (R-STSP). While in the classical STSP a graph is complete and each vertex can be visited at most once, in R-STSP these two constraints are not obligatory which makes the problem more real-life and applicable. To solve the problem, the memetic algorithm (MA) is proposed. After implementing the MA, computer experiments were conducted on the real transport network in Poland. The comparative study of the MA with the genetic algorithm (GA) shows that the MA outperforms the GA.

Keywords: combinatorial optimization, selective travelling salesman problem on a road network, genetic algorithm, memetic algorithm.

1 Introduction

The travelling salesman problem (TSP) is still one of the most challenging combinatorial optimization problems [1]. Given a list of cities and distances between each pair of them, the task is to find a shortest possible tour that visits each city exactly once. Since the TSP is an NP-hard problem [2], it is used as a benchmark for many heuristic algorithms.

So far many versions of the TSP were specified due to various applications in planning, logistic, manufacture of microchips, tourism and others [3], [4]. While in the classical TSP each city must be visited exactly once, some versions of the problem propose to select cities depending on a profit value that is gained when the visit occurs. This class of TSP is called the selective travelling salesman problem (STSP) [5].

The problem which is studied in our paper falls into STSP but its definition introduces two very important modifications. While in the classical STSP a graph is complete and each vertex can be visited at most once, in our approach these two assumptions are not obligatory. Further, not every pair of vertices must be connected with an undirected edge and any vertex in a resultant tour can be multiply visited. The reason of these modifications is that in many real-life applications of TSP it is more appropriate to model a network of connections as an incomplete graph. For example,

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 684–694, 2011.

in transport network roads connecting cities form an incomplete graph. Admittedly, we can transform an incomplete network in a complete one by adding dummy edges, but this transformation can significantly increase a graph density. This, in turn, has a direct impact on increasing of the search space and as a result on the execution time of the algorithm [6]. This issue was considered by Fleischmann [7], who introduced the notion of the Travelling Salesman Problem on a Road Network (R-TSP) and recently by Sharma [8]. An obvious consequence of the assumption of an incomplete graph is a possibility of multiple visits to a vertex. Moreover, in many applications, specially dealing with transport, returns are natural: one may want to travel using repeated fragments of his route. Besides, a possibility of multiply visiting cities enables including hovering vertices to a tour. This version of STSP, with two above mentioned assumptions, will be called the selective travelling salesman problem on a road network (R-STSP).

This paper presents a new memetic algorithm (MA) with local search procedure in a form of heuristic mutation for solving the R-STSP. The term "memetic" comes from the Richard Dawkin's term "meme". The main difference between memes and genes is that memes can be improved by the people. MAs are extensions of standard GAs (or alternative population-based algorithms). They use a separate individual learning or local improvement procedures to improves genotypes. It is this advantage that the MAa have over simple GAs.

The MA described in the paper is the improved version of authors' GA described in [9].

The paper is organized as follows. Section 2 presents the specification of the R-STSP and illustrates it on a simple example. Next section describes the MA with particular focus on a mutation operator. In Section 4 the MA is evaluated and compared with the GA through many experiments on the real transport network in Poland consisting of 306 cities. Finally, the conclusions of this study are drawn and future directions for subsequent research are discussed.

2 Problem Definition

A network of cities in R-STSP can be modeled as a weighted, undirected graph $G = <V, E, d, p>$, where V is a set of vertices (cities), E is a set of edges (transport connections between cities), d is a function of weights (distances between cities) and p is a vector of profits (profits from sale). Each vertex in G corresponds to a given city in a network and is represented by a number from 1 to n, where n is the number of cities in the network. Vertex 1 has a special meaning and is interpreted as the central depot. An undirected edge $\{i, j\} \in E$ is an element of the set E and means that there is a direct two-way road from the city i to the city j. We assume that G is compact but not necessarily complete. The weight d_{ij} for an undirected edge $\{i, j\}$ denotes a distance between cities i and j. Additionally, with each vertex a non-negative number meaning a profit is associated. Let $p=\{p_1, p_2, \ldots, p_n\}$ be a vector of profits for all vertices. Each value of a profit p_i is a positive number. An important assumption is that a profit is realized only during first visiting of a given vertex. At the input of the problem we have: graph G and c_{max} value so that: $G = <V, E, d, p>$ is an undirected compact graph with function of weights d and vector of profits p and c_{max} - a constraint

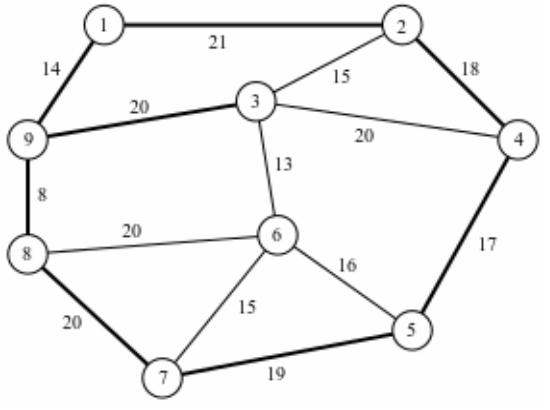


Fig. 1. A graph representation of an exemplary network

for a maximal length of a route. At the output of the problem we obtain route r in graph G, so that: starts and ends in the central depot, length of the route is not greater than c_{max} and total profit of the route is maximal.

A graph representation of an exemplary network of cities is shown in Fig. 1. It is a simple example of the network which is composed of nine cities. The d_{ij} values are marked on the edges and the p_i values are: {5, 5, 3, 5, 5, 2, 5, 5, 5}. One possible solution for this graph for c_{max}=160 can be the cycle r = (1, 2, 4, 5, 7, 8, 9, 3, 9, 1), with the profit equal to 38 and the tour length equal to 157 = 21+18+17+ 19+20+8+20+20+14. This tour is marked in Fig. 1 with bold edges. The graph in Fig. 1 will be used to illustrate all genetic operators described in Section 3.

3 The MA

The first step in adopting the MA for the R-STSP is encoding a solution into a chromosome. Among several different representations for the TSP [10], the path representation is the most natural and is used in the MA for the R-STSP described in the paper. In this approach, a tour is encoded as a sequence of vertices. For example, the tour 1 - 2 - 3 - 9 - 1 is represented by the sequence (1, 2, 3, 9, 1), as was presented in the Section 2.

The block diagram of the MA described in this paper is illustrated in Fig. 2.

The MA starts with a population of P_{size} solutions of R-STSP. The initial population is generated in a special way. At the beginning we randomly choose a vertex v adjacent to the vertex 1 (the start point of the tour). We add the distance d_{1v} to the current tour length. If the current tour length is not greater than $0.5 \cdot c_{max}$ we continue the tour generation, starting now at the vertex v. We again randomly select a vertex u, but this time we exclude from the set of possible vertices the vertex 1 (the next-to-last vertex in the partial tour). This assumption prevents from continual visiting a given vertex but is relaxed if there is no possibility to choose another vertex. If the current tour length exceeds $0.5 \cdot c_{max}$, we reject the last vertex and return to the vertex 1 the same way in reverse order. This strategy insures that the tour length does not exceed

c_{max}. For example, one possible individual for c_{max} = 120 generated according to the above method can be (1, 2, 4, 3, 4, 2, 1) which has the tour length equal to 118. It is easy to observe that such an idea of generating the initial population causes that individuals are symmetrical in respect of the middle vertex in the tour. However, experiments show that the MA quickly removes these symmetries.

The next step is to evaluate individuals in the current population by means of the fitness function. The fitness of a given individual is equal to collected profit under the assumption that a profit is gained only during first visiting of a given vertex. For example, the fitness of the individual represented by the chromosome (1, 2, 4, 3, 4, 2, 1) is equal to 18.

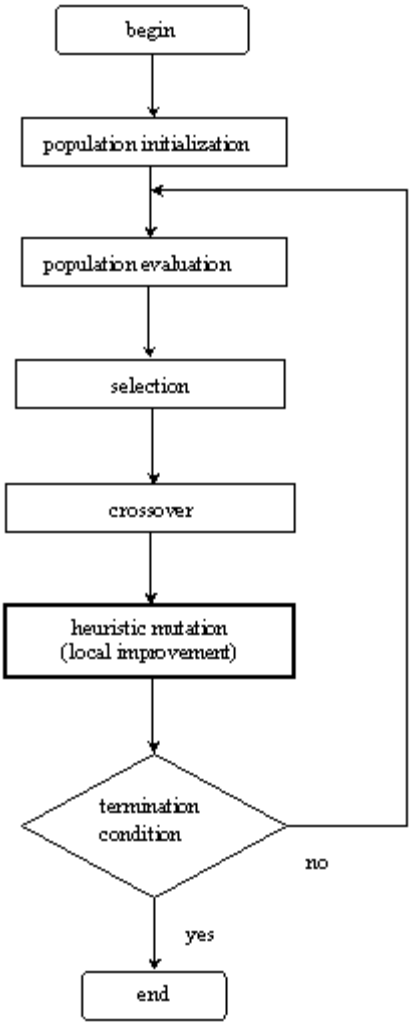


Fig. 2. The block diagram of proposed MA

Once we have the fitness function computed, the MA starts to improve the current population through repetitive application of selection, crossover (also called recombination) and mutation. The MA stops after n_g generations and the result tour is the best individual from the final generation.

In our experiments we use tournament selection: we select t_{size} different individuals from the current population and determine the best one from the group. The winner is copied to the next population and the whole tournament group is returned to the old population. This step is repeated P_{size} times. The parameter t_{size} should be carefully set because the higher t_{size}, the faster convergence of the MA.

Crossover operator is adapted to our problem. Unlike in the TSP, in the R-STSP there is a possibility that parental chromosomes do not have any common gene (with the exception of the first and the last gene). In this situation crossover can not be performed and parents remain unchanged. Otherwise, recombination is conducted in the following way. First we randomly choose one common gene in both parents. This gene will be the crossing point. Then we exchange fragments of tours from the crossing point to the end of the chromosome in two parental individuals. If offspring individuals preserve the constraint c_{max}, they replace in the new population their parents. If one offspring individual does not preserve the constraint c_{max}, its position in the new population is occupied by better (fitter) parent. If both children do not preserve the constraint c_{max}, they are replaced by their parents in the new population. The example of the crossover is presented in Fig. 3 with the assumption that the $c_{max} = 140$.

The length of the tours represented by offsprings are equal to 117 and 106, respectively. Since both offspring individuals preserve the constraint c_{max}, they replace in the new population their parents.

The last genetic operator the population undergo is a mutation. In authors' previous paper [9] an ordinary (classic) mutation was used. It is performed in the following way. First we randomly select a position between the two neighbouring genes in a chromosome. Then we randomly choose a vertex which can be inserted at the selected position. The vertex q can be inserted between two neighbouring genes with values u and v if $\{u, q\} \in E$ and $\{q, v\} \in E$. If inserting a vertex does not cause the exceedance of

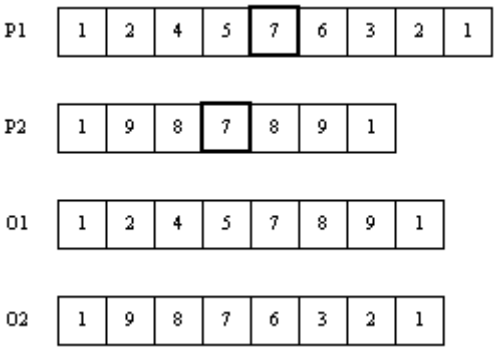


Fig. 3. The example of the crossover operator (crossing point is bolded)

the c_{max}, we keep this new vertex in the tour otherwise we do not insert it. This kind of mutation will be called the classic mutation (CM) throughout the rest of the paper.

Fig. 4 illustrates the example of CM performed on the individual O1 (created during crossover, Fig. 3).

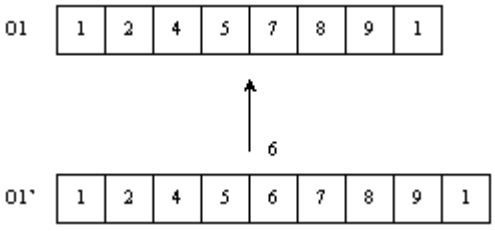


Fig. 4. The example of the CM ($c_{max} = 140$)

The GA presented in [9] worked in the same way as the MA with the only difference in mutation: the GA used CM and the MA uses the heuristic mutation (HM) described below.

The HM is performed as follows. For all neighbouring genes we calculate the set of all possible vertices which can be inserted between a given pair of genes. Next we choose from this set the vertex v which satisfies two conditions: inserting v does not cause the exceedance of the c_{max} and v in not present in the individual being mutated.

If there is more than one vertex which satisfies these conditions, we choose the vertex with the highest profit. Finally, the HM is performed on the best position i.e. between this pair of genes for which inserting v causes the maximal increment of fitness of the individual. Fig. 5 illustrates the example of the HM performed on the individual O1 (created during crossover, Fig. 3). One can see that there are three possibilities of places of inserting a new vertex but the most profitable of them is the vertex 3 ($p_3 = 3$, $p_6 = 2$).

While the CM tries to improve an individual in only one randomly selected position, the HM chooses the best possible position for inserting a vertex. Because of this heuristic of local improvement, the HM has considerable higher probability of improving an individual than the CM. This fact was confirmed by many experimental results described in Section 4.

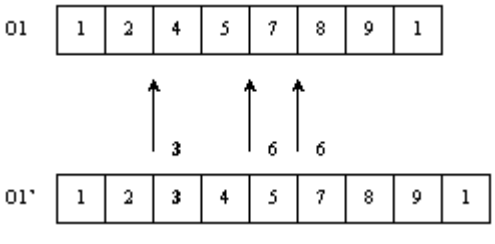


Fig. 5. The example of the HM ($c_{max} = 140$)

4 Experimental Results

We conducted many experiments on the real road network of 306 cities in Poland. The tested data of the network can be found on the website *http://piwonska.pl/research* in two text files: *cities.txt* and *distances.txt*. The network which is written in *distances.txt* file was created from a real map, by including to a graph main segments of roads in the whole Poland. The capital of Poland, Warsaw, was established as the central depot. Profits associated with a given city (written in the file *cities.txt*) were determined according to a number of inhabitants in a given city. The more inhabitants, the higher profit associated with a given city. These rules are presented in a Tab. 1.

Table 1. Rules for profits determining

number of inhabitants	profit
under 10000	1
(10000, 20000]	2
(20000, 30000]	3
(30000, 40000]	4
over 40000	5

In this section we present the comparison results of two approaches: the MA with HM and the GA with CM. In each experiment we set $P_{size} = 300$, $t_{size} = 3$ and $n_g = 100$. These parameters were established through many tests conducted for the GA and described in [9]. Increasing the population size above 300 did not bring significant improvement and was too time consuming. Similarly, iterating the GA over 100 generations did not influence the results: the GA always converged earlier.

Tests were performed for five c_{max} values: 500, 1000, 1500, 2500, 3000. For each c_{max} we run both algorithms, implemented in C language, ten times on Intel (R) Core TM2 Duo CPU T8100 2,1 GHZ. Results of experiments are presented in Tab. 2 and Tab. 3.

Table 2. The best results from ten runs of the GA and the MA

	GA		MA	
c_{max}	profit	length	profit	length
500	87	367	114	428
1000	130	982	176	998
1500	173	1483	240	1452
2000	235	1953	279	1980
2500	279	2485	338	2476
3000	304	2880	366	2998

One can see that the best as well as the average profits are much better in the case of the MA than in the GA. In case of the best results (Tab. 2) the average improvement rate of the profit is equal to 27,6% for all c_{max} values. Similar improvement can be observed for the average results (Tab. 3) and is equal to 22,2%. It is interesting that the resultant tours (the best as well as the average ones) have comparable lengths. This fact can be explained that any of the mutations does not consider the increment of the resultant tour length as a criterion of inserting a vertex. They only assure that the c_{max} constraint will not be violated.

Table 3. The average values from ten runs of the GA and the MA

	GA		MA	
$_{max}$	profit	length	profit	length
500	81,7	445	101	459,7
1000	120,4	989,5	145,3	981,6
1500	151,8	1481,2	215,1	1481,1
2000	220,2	1947,6	261,4	1943,4
2500	260,2	2488,4	297,6	2472,1
3000	293	2912,8	333,9	2987,9

Another important improvement concerns the convergence of both algorithms. The MA finds the best individual faster than the GA and these differences in the velocity of the convergence intensify with the c_{max} increase. Fig. 6 and 7 show this effect for two extreme values of c_{max}.

One can see that the length of the chromosome coding the best individual in the case of the MA is greater than in the GA. This is due to the fact that the HM has greater probability of adding a city to a chromosome. It is worth to mention that in resultant chromosomes one can observe some number of repeated cities. However the number of returns in a given chromosome is relatively small. It is the desirable effect because a repeated city in a tour does not increase the fitness of a tour.

Tab. 4 presents the average execution time (in ms) of both algorithms. As it was expected, the MA runs longer than the GA but these differences are not considerable. This effect was obtained due to the efficient implementation of the HM. For all

Table 4. Average execution time in milliseconds of the GA and the MA

cmax	GA	MA
500	22574	29460
1000	45293	65095
1500	49539	69442
2000	60877	70820
2500	62231	73825
3000	62820	74444

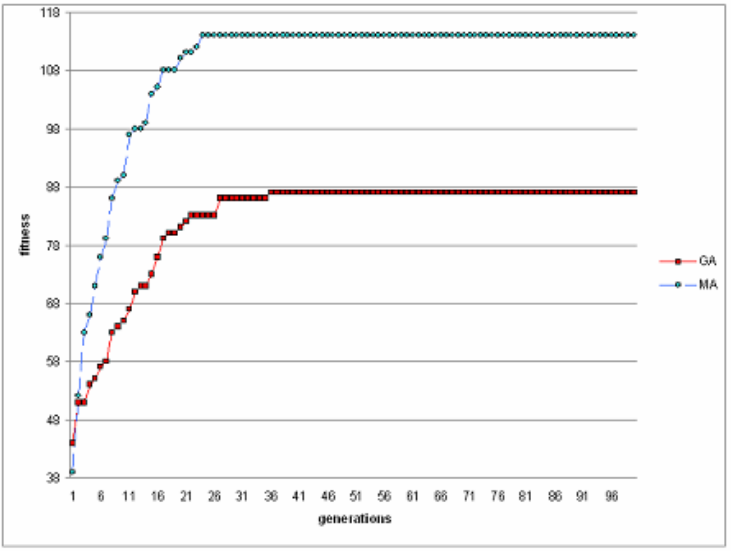


Fig. 6. The best runs of the GA and the MA for $c_{max} = 500$

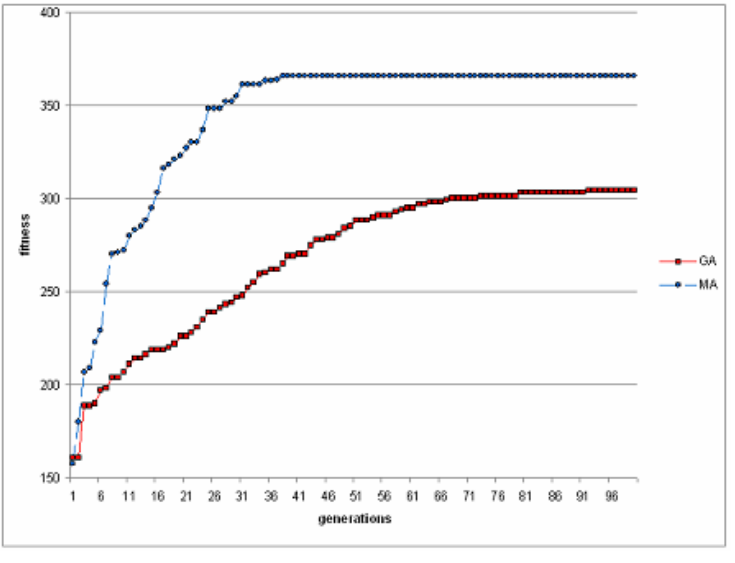


Fig. 7. The best runs of the GA and the MA for $c_{max} = 3000$

neighbouring cities we calculated the set of all possible cities which could be inserted between a given pair of cities in the precomputation phase of the algorithm (before the MA starts). Due to this precomputation the process of determining the best profitable city for a given pair of cities can be done in a constant time. As a result, the time complexity of a single run of the HM is lineary dependent on the chromosome length.

5 Conclusions

The paper presented the R-STSP which is more applicable than standard STSP because of two important assumptions: an incompleteness of a network and a possibility of returns to the same city. The authors proposed the memetic algorithm with a heuristic mutation operator for solving this problem. The method was verified by testing on the real network including main roads and cities in Poland. The results were compared with the results obtained by the genetic algorithm [9]. Conducted tests showed that the MA worked much better than the GA, taking into account the quality of obtained tour as well as the convergence of the fitness function to the optimum. Moreover, this improvement was achieved only by the small growth of the time complexity of the MA. This effect was reached due to the efficient implementation of the HM which used the precomputed sets of all possible cities.

Future research will be conducted in several ways. We plan to add to the HM another condition of choosing a vertex, taking into account the increment of the tour length. Since the problem considered in this paper is new, we also plan to compare the MA for the R-STSP with another heuristic approaches such as tabu search or ant colony optimization.

The problem considered in the paper can be extended to time-dependent variant in which the costs of travel between cities depend on the starting time of a journey. Such a version of the R-STSP can be applied in tourist planners [11] and Intelligent Transportation Systems [12]. The problem can be also extended to its asymmetric version in which distances between cities *i* and *j* are not equal for both directions [13].

References

1. Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton (2006)
2. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1979)
3. Liu, S., Pinto, J.M., Papageorgiou, L.G.: A TSP-based MILP Model for Medium-Term Planning of Single-Stage Continuous Multiproduct Plants. Industrial & Engineering Chemistry Research 47(20), 7733–7743 (2008)
4. Voelkel, T., Weber, G.: Routecheckr: personalized multicriteria routing for mobility impaired pedestrians. In: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 185–192 (2008)
5. Feillet, D., Dejax, P., Gendreau, M.: Traveling Salesman Problems with Profits. Transportation Science 39(2), 188–205 (2005)
6. Rasmussen, R.: TSP in spreadsheets – A fast and flexible tool. Omega 39, 51–63 (2011)
7. Fleischmann, B.: A new class of cutting planes for the symmetric travelling salesman problem. Mathematical Programming 40, 225–246 (1988)
8. Sharma, O., Mioc, D., Anton, F., Dharmaraj, G.: Traveling salesperson approximation algorithm for real road networks. In: ISPRS WG II/1,2,7, VII/6 International Symposium on Spatio-temporal Modeling, Spatial Reasoning, Spatial Analysis, Data Mining & Data Fusion (STM 2005), Beijing, China (2005)
9. Koszelew, J., Piwonska, A.: Tunning Parameters of Evolutionary Algorithm in Travelling Salesman Problem with Profits and Returns. Archives of Transport System Telematics 39(1), 17–22 (2010)

10. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1999)
11. Ludwig, B., Zenker, B., Schrader, J.: Recommendation of Personalized Routes with Public Transport Connections. In: Intelligent Interactive Assistance and Mobile Multimedia Computing. Communications in Computer and Information Science, vol. 53, Part 3, pp. 97–107 (2009)
12. Miller, J., Sun-il, K., Menard, T.: Intelligent Transportation Systems Traveling Salesman Problem (ITS-TSP) - A Specialized TSP with Dynamic Edge Weights and Intermediate Cities. In: 13th IEEE Intelligent Transportation Systems Conference, Madeira Island, Portugal, pp. 992–997 (2010)
13. Ascheuer, N., Fischetti, M., Grötschel, M.: Solving the Asymmetric Travelling Salesman Problem with Time Windows by Branch-and-Cut. Mathematical Programming 90(3), 475–506 (2001)

Application of DRSA-ANN Classifier in Computational Stylistics

Urszula Stańczyk

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

Abstract. Computational stylistics or stylometry deals with characteristics of writing styles. It assumes that each author expresses themselves in such an individual way that a writing style can be uniquely defined and described by some quantifiable measures. With help of contemporary computers the stylometric tasks of author characterisation, comparison, and attribution can be implemented using either some statistic-oriented approaches or methodologies from artificial intelligence domain. The paper presents results of research on an application of a hybrid classifier, combining Dominance-based Rough Set Approach and Artificial Neural Networks, within the task of authorship attribution for literary texts. The performance of the classifier is observed while exploiting an analysis of characteristic features basing on the cardinalities of relative reducts found within rough set processing.

Keywords: Classifier, DRSA, ANN, Computational Stylistics, Characteristic Feature, Relative Reduct, Authorship Attribution.

1 Introduction

Computational stylistics or stylometry is typically employed to prove or disprove authenticity of documents, to establish authorship in cases when the author is either unknown or disputed, to detect plagiarism, for automatic text categorisation. These aims can be achieved by exploiting the fundamental concept of *writer invariant*, such a set of numerical characteristics which capture the uniqueness and individuality of a writing style [1].

The characteristics used should, on one hand, enable distinguishing an author from others thus allowing for classification and recognition, but on the other hand, they should prevent easy imitation of someone else's style. Therefore the textual descriptors employed usually reflect rather subtle elements of style, employed subconsciously by the authors, such as frequencies of usage for letters or words (lexical descriptors), the structure of sentences formed by the punctuation marks (syntactic descriptors), the organisation of a text into headings, paragraphs (structural markers), or exploited words of specific meaning (content-specific markers) [8].

The selection of textual descriptors is one of crucial decisions to be made within the stylometric processing, while the second is the choice of methodology

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 695–704, 2011.

to be employed. While one path leads to statistics, the other exploits techniques from artificial intelligence area that perform well in cases with knowledge uncertain and incomplete. Dominance-based Rough Set Approach (DRSA) and Artificial Neural Networks (ANN) belong in this latter category. Both methodologies can be employed on their own in authorship attribution studies [11,13], yet a combination of elements of these two yields a hybrid solution that can also be used. Rough set perspective on attributes imposed on characteristic features of ANN classifier brings observations on significance of individual features in the process of classification and recognition and can be exploited for feature selection and reduction [5].

In the past research [10] there was performed an analysis of characteristic features for ANN classifier basing on the concept of a relative reduct in the discrete case of Classical Rough Set Approach (CRSA), as defined by Z. Pawlak [7]. Yet discretisation means discarding some information so in the later research the analysis was based on relative reducts and decision rules calculated within DRSA methodology that allows for ordinal classification [3,4,9]. These studies concerned the frequency of usage of features in calculated reducts and rules [12].

The paper presents the research that is a continuation of experiments on combining elements of rough set theory and artificial neural networks applied in authorship attribution. The tests presented involved more detailed analysis of relative reducts found, by observing their cardinalities and how they relate to individual attributes. The results confirm the findings from the past and indicate that if there are considered characteristic features most often appearing in relative reducts with lowest cardinalities, these features can constitute some proper subsets of features which still preserve the classification accuracy of ANN with the full set of features, or even increase this accuracy.

2 Computational Stylistics

Stylometric processing requires such analysis of texts that yields their unique characteristics, which allows for characterisation of authors, finding similarities and differences, and attributing the authorship [2].

In computational stylistics community there is no consensus which characteristics should be used and sets of textual descriptors giving the best results are to high degree task-dependent. Therefore it is quite common that the whole stylometric processing is divided into several steps. In the initial phase the choice of features is arbitrary and rather excessive than minimal. Next phases of processing require establishing the significance of individual features and possibly discarding some of them. Thus in fact the reduction of features is considered not in the stylometric context, but from the point of view of the processing methodology employed.

Characteristics must be based on some sufficiently wide corpus of texts. Very short text samples can vary to such extent, that any conclusions from them cannot be treated as reliable. To satisfy this requirement for the experiments as the input data there were chosen works of Henry James and Thomas Hardy, two

famous writers from XIXth century. The samples were created by computing characteristics for markers within parts taken from selected novels. The fragments of texts were of approximately the same length, typically corresponding to chapters. For the learning set (total of 180 samples) there were 30 parts from 3 novels for each writer. For the testing set (80 samples) there were 8 parts from other 5 novels.

The base set of textual descriptors (total of 25) was built with frequencies of usage for the arbitrarily selected 17 common function words and 8 punctuation marks, as follows: but, and, not, in, with, on, at, of, this, as, that, what, from, by, for, to, if, a fullstop, a comma, a question mark, an exclamation mark, a semicolon, a colon, a bracket, a hyphen. It is assumed that whenever there is the left bracket, the right one always eventually follows, thus this is considered as a single instance.

3 Classification with ANN

Construction of a connectionist classifier involves decision as to the network topology and the one selected for experiments was Multilayer Perceptron [13], a unidirectional, feedforward network, with neurons grouped into some number of layers, implemented with California Scientific Brainmaker simulation software. There was assumed sigmoid activation function for neurons, and as the training rule there was employed classical backpropagation algorithm, which minimises the error on the network output,

$$e(\mathbf{W}) = \frac{1}{2} \sum_{m=1}^{M} \sum_{i=1}^{I} \left(d_i^m - y_i^m(\mathbf{W})\right)^2 \tag{1}$$

which is a sum of errors on all I output neurons for all M learning facts, each equal to the difference between the expected outcome d_i^m and the one generated by the network $y_i^m(\mathbf{W})$, for a current vector of weights ($\mathbf{W}$) associated with interconnections.

The number of network input nodes corresponds to the number of characteristic features considered for the task, while the network outputs typically reflect recognition classes. In the experiments the first parameter was initially equal 25, then decreased when the analysis of features was exploited in feature reduction, and the second was always two, for two recognised authors.

There are many rules for a recommended number of hidden layers and neurons in them, yet as the resulting performance is task-dependent, by tests it was established that the structure with the highest classification accuracy for the initial set of features was the one with two hidden layers, the first layer with $\lceil 3/4$ number of inputs$\rceil$ neurons, the second with $\lfloor 1/4$ number of inputs$\rfloor$.

To minimise the influence of the initiation of weights on a network training process, there was implemented multi-starting approach: for each network configuration the training was performed 20 times. Basing on such series there was obtained the worst, best, and average performance and only this last is presented in the paper. For 25 features the average classification accuracy was 82.5%.

4 Analysis of Characteristic Features Based on Relative Reducts

In Classical Rough Set Approach the granules of knowledge observed are the equivalence classes of objects that cannot be discerned with respect to a set of considered criteria [7]. This allows only for nominal classification. Dominance-based Rough Set Approach, proposed to deal with multi-criteria decision making problems [4,9], substitutes the indiscernibility relation with dominance, and assumes that for all attributes there is present some preference order in their value sets. The granules of knowledge become dominance cones. This enables ordinal classification and processing of real-valued input data sets.

In rough set theory relative reducts are such irreducible subsets of condition attributes that preserve the quality of approximation of a decision table with respect to the selected criteria. The decision table can have many reducts and their intersection is called a core [6]. If an attribute belongs to the core, it is necessary for classification. When the core is empty, then all attributes can be treated as equally good choices, as long as their subset is one of reducts.

Before relative reducts can be found, firstly the decision table must be constructed. In the experiments it was based on the same set of learning samples as for ANN classifier. The columns of condition attributes corresponded to characteristic features previously defined, while the single decision attribute assumed values reflecting two recognised classes. DRSA methodology requires also a preference order to be specified for all attributes.

When term frequencies are used as the characteristic features for the constructed rule-based classifier, there is no doubt that the values observed are ordered, yet their preference cannot be established within stylometric domain, as some a priori, universal knowledge about the frequencies with reference to particular authors does not exist. However, basing on them we can determine authorship, thus it is reasonable to expect that such preference does exist, that observation of certain, lower or higher, frequencies is characteristic for specific authors. That is why the preference order is either assumed arbitrarily or found in an experimental way. In the research the preference was assumed arbitrarily.

Analysis of the decision table yielded 6664 relative reducts, with cardinalities varying from 4 to 14. The core turned out to be empty, while the union of all relative reducts gave the set of all condition attributes.

Relative reducts are characterised by their cardinalities and by attributes used in their construction. This perspective can be reversed, that is, the attributes can be perceived by the reducts they belong to, as presented in Table 1. Similar perspective was exploited in the past research [12,11] within the reduction of features for both rule-based and connectionist classifiers, yet the tests performed so far concerned only the frequency of usage in relative reducts and decision rules for individual attributes.

The choice of a reduct within rough set approach can be dictated basing on its cardinality. Fewer attributes means less processing (and fewer rules if a decision algorithm is constructed), however, it should be remembered that the patterns of input data, which relative reducts point, are detected in the decision table, that

Table 1. Analysis of characteristic features based on relative reducts

	Reduct cardinalities										
	4	5	6	7	8	9	10	11	12	13	14
Number of reducts	6	160	635	1323	1210	1220	1245	582	233	46	4
Attribute	Number of reducts of specific cardinality for attributes										
of	0	16	177	580	625	771	775	409	103	19	3
.	3	74	226	477	560	696	729	300	108	16	1
on	1	9	184	400	488	631	808	364	157	38	3
,	0	5	70	308	423	707	811	409	174	32	4
not	4	53	332	579	494	518	512	233	49	4	0
;	0	3	87	318	517	644	658	326	157	28	2
in	2	50	165	434	374	519	742	305	120	15	0
by	3	108	228	461	581	598	479	138	47	5	0
this	0	15	117	348	457	549	555	331	169	41	3
at	0	12	118	429	477	466	606	302	146	28	1
to	0	18	103	427	446	466	605	279	125	27	1
:	1	23	130	351	342	522	602	265	116	29	3
!	0	39	183	396	489	494	483	221	52	9	2
and	6	144	566	969	498	116	22	3	0	0	0
from	3	90	186	398	421	460	337	207	132	36	3
with	0	6	102	355	337	382	482	299	157	39	2
as	0	14	137	355	288	366	498	278	141	28	3
-	0	14	93	316	405	372	415	241	144	32	3
?	0	26	124	212	260	275	379	283	121	28	4
for	0	8	74	247	333	276	364	195	79	29	4
if	1	29	138	265	141	247	344	299	96	22	2
what	0	11	124	203	204	226	355	176	95	19	2
(	0	10	44	152	179	261	329	218	158	41	3
that	0	14	69	160	180	274	337	189	93	23	4
but	0	9	33	121	161	144	223	132	57	10	3

is ensuring classification for the learning samples, whereas in the testing set these patterns may be present to some degree only. Thus the minimal cardinality of selected reducts does not guarantee the highest classification accuracy for a rule-based classifier. Neither do the relative reducts by themselves perform well when directly applied as feature selectors for a connectionist classifier [10]. However, it can be reasonably expected that the importance of features is related to the cardinalities of reducts they are included in.

It could be argued that attributes belonging to relative reducts with low cardinalities are the most important as only few are enough to ensure the same quality of approximation as the complete set of attributes. On the other hand, higher cardinality can be treated as keeping some bigger margin for possible error, or allowing for bigger difference between the learning and the testing set. The question which of these two approaches brings better results, understood as higher classification accuracy, when transfered into the context of data processing with ANN, needs to be verified with tests.

5 Experiments Performed

In order to observe the influence of the individual characteristic features on the performance of ANN classifier, when these features are seen from DRSA relative reducts perspective, three orderings of the studied features were assumed, as given in Table 2.

For Order 1 and Order 2 there were taken into account all relative reducts the attributes belong to. Order 1 is based on the value of the quality indicator calculated as the sum of numbers of attribute occurrences in reducts of specific cardinality divided by these cardinalities. Such calculation assumes higher influence of smaller reducts. For Order 2 the quality indicator specified equals the sum of products: reduct cardinality multiplied by the number of occurrences in such reducts for all attributes. This approach assigns higher significance to bigger reducts. And finally for Order 3 there were taken into account only the frequencies of usage in the smallest reducts (with cardinality 4 and 5), disregarding reducts with higher cardinalities.

As shown in Table 1, there were relatively few small and large reducts. The majority of reducts had cardinalities from 7 to 10. This resulted in some simi-

Table 2. Ordering of characteristic features based on an analysis of cardinalities of relative reducts in which the features are included

Order 1			Order 2			Order 3				
of	404.29		of	30915		and	6	144		
.	379.17	M1	on	28040	M1	not	4	53	M1	
on	351.08	M2	.	27929		by	3	108	M2	
not	345.73		,	27517		from	3	90		
and	340.67		;	25137	M2	.	3	74		
by	330.02	M3	in	24359	L7	in	2	50	M3	L7
,	323.63		this	23604	M3	if	1	29	M4	
in	317.50	L7	at	23293		:	1	23		
;	307.53	M4 L6	not	23263	L6	on	1	9		L6
at	297.20	M5	to	22443	M4	!	0	39	M5	L5
this	293.38		by	22114		?	0	26	M6	L4
to	287.72		:	21536	L5	to	0	18	M7	
!	284.44	L5	!	20453	M5	of	0	16		
from	276.84	M6	with	19789		this	0	15		L3
:	273.81	L4	from	19613		-	0	14	M8	
with	245.09	M7	as	19111		that	0	14		
as	242.21		-	18415	L4	as	0	14		L2
-	233.49	L3	and	16204	M6	at	0	12	M9	
?	195.36	M8	?	15688	L3	what	0	11		
for	184.74		for	14527	M7	(	0	10		L1
if	183.39	L2	if	14378	L2	but	0	9		
what	163.50		(	13318		for	0	8		
(	151.68		what	12787		with	0	6		
that	150.79	L1	that	12430	L1	,	0	5		
but	100.74		but	8212		;	0	3		

larities that can be observed between orderings presented in Table 2. Yet there are also sufficient differences to give motivation for testing all three.

For each of the three orderings the condition attributes can be considered and subsequently reduced from either more or less significant side, with the highest or lowest values of the calculated measures, which means three groups of tests, each further divided into two subgroups. The series keeping more important features and discarding less important were labelled "L", while these reducing more important features and keeping less important were denoted with "M".

Fig. 1 shows that for both Order 1 and Order 2 discarding the features that are considered less significant causes at first just a slight, then noticeable decrease in the classifier performance. The results are better when the reduction of features is based on Order 1, assuming greater significance of smaller reducts, which yields the conclusion that the presence or absence of attributes rarely used in small reducts is not very important for the classification.

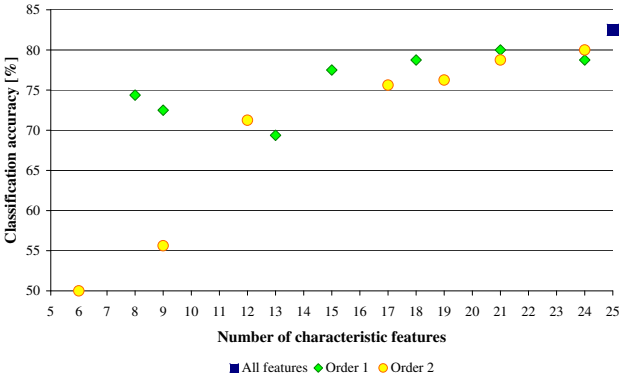


Fig. 1. ANN classification accuracy in relation to the number of features, with their reduction from less significant side for Order 1 and Order 2

Fig. 2 indicates that when more important features are removed and less important kept the results for both orders are quite similar. For Order 1 the same classification as for the whole set of features can be obtained for 66.66% inputs left. Removing more important features in Order 2 means discarding these attributes that are most often present in reducts with high cardinalities and when 20% of inputs are reduced the classification slightly increases to 83.75%, then with further reduction gradually decreases.

The performance of the classifier observed in reduction of characteristic features according to Order 1 and 2 results in the conclusion that when cardinalities of relative reducts are considered as importance indicators for condition attributes, then smaller reducts seem to be more informative.

This leads to the third ordering of features and the performance of the classifier as presented in Fig. 3. With focus on only the smallest reducts, with cardinalities 4 and 5, removing more important features while keeping these less important

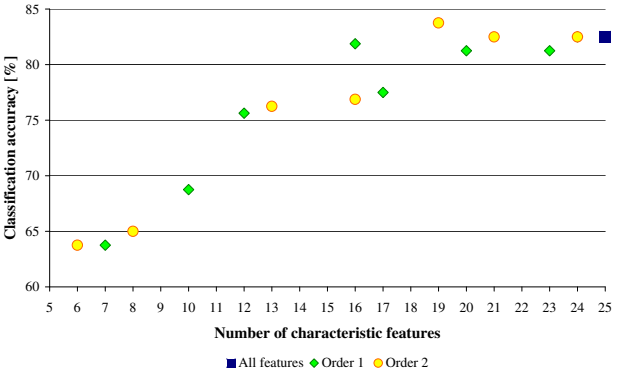


Fig. 2. ANN classification accuracy in relation to the number of features, with reduction from more significant side for Order 1 and Order 2

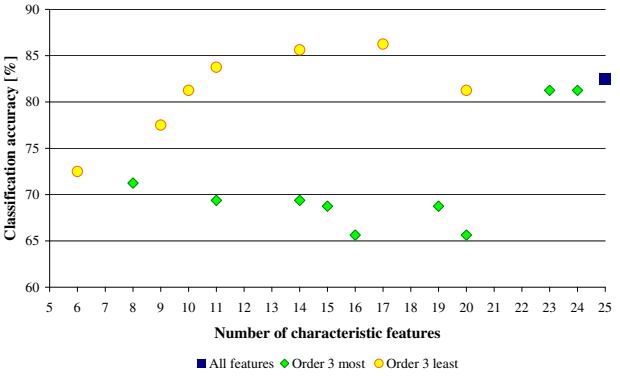


Fig. 3. ANN classification accuracy in relation to the number of features, reduction for Order 3 (focus on only the smallest reducts) from both sides

ones gives the satisfactory classification accuracy even when there are more than 50% of characteristic features reduced. Within all three orderings of features this is the only one that enables for such a significant reduction while not diminishing the power of ANN classifier.

It should be noticed that in all these tests, for all three orderings of considered attributes, the performance of the classifier is always at least slightly decreased when removing less significant features. On the other hand, reduction of more significant attributes yields several subsets of attributes for which the classification is the same as for the whole set or increased.

This could be considered as counter-intuitive that features considered as more important in the rough set perspective are less important for ANN classifier. Yet the rule-based methodology focuses on dominant patterns observed in the training samples while a connectionist approach looks for subtle differences, hence

it is unavoidable that they assume different levels of importance for individual features and makes a combination of both all the more interesting.

These observations confirm findings from the past research [10] and bring the conclusion that even though relative reducts found in DRSA processing just by themselves do not perform well enough as feature selectors for a connectionist classifier, the analysis of their cardinalities, which results in ordering of attributes reflecting their frequency of usage, still can be used to advantage in feature selection and reduction process.

6 Conclusions

The paper presents results of experiments concerning an application of a hybrid classifier in the computational stylistics task of authorship attribution, constituting another step within the research track continued over some past years.

The data processing described was performed in three stages. Firstly, there were selected some characteristic features that gave satisfactory classification accuracy of an artificial neural network. Secondly for this set of features there were calculated relative reducts as defined by Dominance-based Rough Set Approach. Relative reducts are characterised by their cardinalities and the attributes they include. Reversing this perspective, the attributes were ordered reflecting how often they were used in construction of reducts with various cardinalities. Basing on these orderings of features in the third step there was conducted reduction of features while observing how this influences the classifier performance.

The experiments show that greater importance in the classification process executed by ANN classifier was associated with these features that were perceived as less important from relative reduct perspective. The presence or absence of the features considered as more significant influences the power of the classifier in smaller degree. This observation can be considered as counter-intuitive, yet it actually shows the opposite attitudes of the two methodologies employed: rule-based approach looks for dominant features and patterns, while connectionist approach relies on detecting rather subtle and less obvious relationships amongst input data.

Acknowledgments. 4eMka Software used in search for relative reducts [4,3] was downloaded in 2008 from the website of Laboratory of Intelligent Decision Support Systems, (http://www-idss.cs.put.poznan.pl/), Poznan University of Technology, Poland.

References

1. Burrows, J.: Textual analysis. In: Schreibman, S., Siemens, R., Unsworth, J. (eds.) A companion to Digital Humanities, ch. 23, Blackwell, Oxford (2004)
2. Craig, H.: Stylistic analysis and authorship studies. In: Schreibman, S., Siemens, R., Unsworth, J. (eds.) A companion to digital humanities. Blackwell, Oxford (2004)

3. Greco, S., Matarazzo, B., Slowinski, R.: The use of rough sets and fuzzy sets in Multi Criteria Decision Making. In: Gal, T., Hanne, T., Stewart, T. (eds.) Advances in Multiple Criteria Decision Making, ch. 14, pp. 14.1–14.59. Kluwer Academic Publishers, Dordrecht (1999)
4. Greco, S., Matarazzo, B., Slowinski, R.: Dominance-based rough set approach as a proper way of handling graduality in rough set theory. Transactions on Rough Sets 7, 36–52 (2007)
5. Jelonek, J., Krawiec, K., Slowinski, R.: Rough set reduction of attributes and their domains for neural networks. Computational Intelligence 11(2), 339–347 (1995)
6. Moshkov, 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)
7. Pawlak, Z.: Rough sets and intelligent data analysis. Information Sciences 147, 1–12 (2002)
8. Peng, R., Hengartner, H.: Quantitative analysis of literary styles. The American Statistician 56(3), 15–38 (2002)
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. Stańczyk, U.: Relative reduct-based selection of features for ANN classifier. In: Cyran, K., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. AISC, vol. 59, pp. 335–344. Springer, Heidelberg (2009)
11. Stańczyk, U.: DRSA decision algorithm analysis in stylometric processing of literary texts. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 600–609. Springer, Heidelberg (2010)
12. Stańczyk, U.: Rough set-based analysis of characteristic features for ANN classifier. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 565–572. Springer, Heidelberg (2010)
13. Waugh, S., Adams, A., Twedeedie, F.: Computational stylistics using artificial neural networks. Literary and Linguistic Computing 15(2), 187–198 (2000)

Investigating the Effectiveness of Thesaurus Generated Using Tolerance Rough Set Model

Gloria Virginia and Hung Son Nguyen

University of Warsaw, Faculty of Mathematics, Informatics and Mechanics
Banacha 2, 02-097 Warsaw, Poland

Abstract. We considered the tolerance matrix generated using tolerance rough set model as a kind of an associative thesaurus. The effectiveness of the thesaurus was measured using performance measures commonly used in information retrieval, recall and precision, where they were used for the *terms* rather than *documents*. A corpus consists of keywords defined as highly related with particular topic by human experts become the ground truth of this study. Analysis was conducted based on comparison values of all available sets created. Above all findings, this paper was thought as the fundamental basis that generating an automatic thesaurus using rough sets theory is a promising way. We also mentioned some directions for future study.

Keywords: rough sets, tolerance rough set model, thesaurus.

1 Introduction

Rough set theory is a mathematical approach to vagueness [12] that was introduced by Pawlak in 1982 [11]. It's relationship with other approaches has been studied for years and it has been successfully implemented in numerous areas of real-life applications [5]. Tolerance rough set model (TRSM) is one of its extension developed by Kawasaki, Nguyen, and Ho [4] based on the *generalized approximation space* as a tool to model document-term relation in text mining.

Hierarchical and non-hierarchical document clustering based on TRSM has been studied in [4] and [8] respectively and showed that the clustering algorithm being proposed could be well adapted to text mining. The study of TRSM implementation to search results clustering in [8] yielded a design of a Tolerance Rough Set Clustering (TRC) algorithm for web search results and proved that the new representation created had positive effects on clustering quality. For query expansion, the result of TRSM implementation showed that the approach was effective and high search precision was gained [3,8].

The potential of TRSM in automatic thesaurus construction has been revealed in [16]. By employing similar framework of study, this paper presents our investigation on the effectiveness of the thesaurus automatically created, both with and without stemming task on the process. The effectiveness of the thesaurus was calculated using performance measures commonly used in information retrieval which are recall and precision.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 705–714, 2011.

Brief explanation about rough sets theory, generalized approximation space and tolerance rough set model are presented on the next section and then followed by description of data and methodology used in the study. We report and discuss our findings in section 5.

2 Basic Notions on Tolerance Rough Set Model (TRSM)

Rough set theory was originally developed [12] as a tool for data analysis and classification. It has been successfully applied in various tasks, such as feature selection/extraction, rule synthesis and classification [5,10]. The central point of rough set theory is based on the fact that any concept (a subset of a given universe) can be approximated by its *lower* and *upper approximation*.

The classical rough set theory is based on equivalence relation that divides the universe of objects into disjoint classes. For some practical applications, the requirement for equivalent relation has showed to be too strict. The nature of the concepts in many domains are imprecise and can be overlapped additionally.

In [13], Skowron and Stepaniuk introduced a generalized approximation space (GAS) by relaxing the equivalence relation in classical rough sets to a tolerance relation, where transitivity property is not required. Formally, the generalized approximation space is defined as a quadruple $\mathcal{A} = (U, I, \nu, P)$, where

U is a non-empty universe of objects; let $\mathcal{P}(U)$ denote the power set of U,
$I : U \to \mathcal{P}(U)$ is an *uncertainty function* satisfying conditions: (1) $x \in I(x)$ for $x \in U$, and (2) $y \in I(x) \iff x \in I(y)$ for any $x, y \in U$. Thus the relation $xRy \iff y \in I(x)$ is a tolerance relation and $I(x)$ is a tolerance class of x,
$\nu : \mathcal{P}(U) \times \mathcal{P}(U) \to [0, 1]$ is a *vague inclusion function*, which measures the degree of inclusion between two sets. The function ν must be *monotone* w.r.t the second argument, i.e., if $Y \subseteq Z$ then $\nu(X, Y) \leq \nu(X, Z)$ for $X, Y, Z \subseteq U$,
$P : I(U) \to \{0, 1\}$ is a *structurality function*.

Together with uncertainty function I, vague inclusion function ν defines the *rough membership function* for $x \in U, X \subseteq U$ by $\mu_{I,\nu}(x, X) = \nu(I(x), X)$. Lower and upper approximations of any $X \subseteq U$ in $\mathcal{A}$, denoted by $\mathbf{L}_{\mathcal{A}}(X)$ and $\mathbf{U}_{\mathcal{A}}(X)$, are respectively defined as $\mathbf{L}_{\mathcal{A}}(X) = \{x \in U : P(I(x)) = 1 \wedge \nu(I(x), X) = 1\}$ and $\mathbf{U}_{\mathcal{A}}(X) = \{x \in U : P(I(x)) = 1 \wedge \nu(I(x), X) > 0\}$.

Let us notice that the classical rough sets theory is a special case of GAS. However, with given definition above, generalized approximation spaces can be used in any application where I, ν and P are appropriately determined.

Tolerance Rough Set Model (TRSM) [4] was developed as basis to model documents and terms in information retrieval, text mining, etc. With its ability to deal with vagueness and fuzziness, tolerance rough set seems to be promising tool to model relations between terms and documents. In many information retrieval problems, especially in document clustering, defining the similarity relation between document-document, term-term or term-document is essential.

Let $D = \{d_1, \ldots, d_N\}$ be a corpus of documents and $T = \{t_1, \ldots, t_M\}$ set of *index terms* for D. With the adoption of Vector Space Model [7], each document

d_i is represented by a weight vector $[w_{i1}, \ldots, w_{iM}]$ where w_{ij} denoted the weight of term t_j in document d_i. TRSM is an approximation space $\mathcal{R} = (T, I_\theta, \nu, P)$ determined over the set of terms T as follows:

Uncertainty function: $I_\theta(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \theta\} \cup \{t_i\}$, where θ is a positive parameter and $f_D(t_i, t_j)$ denotes the number of documents in D that contain both terms t_i and t_j. The set $I_\theta(t_i)$ is called the *tolerance class* of term t_i,

Vague inclusion function: is defined as $\nu(X, Y) = \frac{|X \cap Y|}{|X|}$,

Structural function: $P(I_\theta(t_i)) = 1$ for all $t_i \in T$.

The membership function μ for $t_i \in T$, $X \subseteq T$ is then defined as $\mu(t_i, X) = \nu(I_\theta(t_i), X) = \dfrac{|I_\theta(t_i) \cap X|}{|I_\theta(t_i)|}$ and the lower, upper approximations and boundary regions of any subset $X \subseteq T$ can be determined – with the obtained tolerance $\mathcal{R} = (T, I, \nu, P)$ – in the standard way, i.e.,

$$L_{\mathcal{R}}(X) = \{t_i \in T \mid \nu(I_\theta(t_i), X) = 1\} \quad . \tag{1}$$

$$U_{\mathcal{R}}(X) = \{t_i \in T \mid \nu(I_\theta(t_i), X) > 0\} \quad . \tag{2}$$

$$BN_{\mathcal{R}}(X) = U_{\mathcal{R}}(X) - L_{\mathcal{R}}(X) \quad . \tag{3}$$

In the context of information retrieval, tolerance class $I_\theta(t_i)$ represents the concept related to t_i. By varying the threshold θ, one can tune the preciseness of the concept represented by a tolerance class. For any set of terms X, the upper approximation $\mathbf{U}_{\mathcal{R}}(X)$ is the set of concepts that share some semantic meanings with X, while $\mathbf{L}_{\mathcal{R}}(X)$ is a "core" concept of X. The application of TRSM in document clustering was proposed as a way to enrich document and cluster representation with the hope of increasing clustering performance.

Enriching document representation: With TRSM, the "richer" representation of document $d_i \in D$ is achieved by simply representing document with its upper approximation, i.e. $\mathbf{U}_{\mathcal{R}}(d_i) = \{t_i \in T \mid \nu(I_\theta(t_i), d_i) > 0\}$

Extended weighting scheme: In order to employ approximations for document, the weighting scheme need to be extended to handle terms that occurs in document's upper approximation but not in the document itself. The extended weighting scheme is defined from the standard TF*IDF by:

$$w^*_{ij} = \frac{1}{S} \begin{cases} (1 + \log f_{d_i}(t_j)) \log \frac{N}{f_D(t_j)} & \text{if } t_j \in d_i \\ \min_{t_k \in d_i} w_{ik} \frac{\log \frac{N}{f_D(t_j)}}{1 + \log \frac{N}{f_D(t_j)}} & \text{otherwise} \end{cases}$$

where S is a normalization factor.

The use of upper approximation in similarity calculation to reduce the number of zero-valued similarities is the main advantage TRSM-based algorithms claimed to have over traditional approaches. This makes the situation, in which two documents have a non-zero similarity although they do not share any terms, possible.

3 Data of the Study

This study used two corpora: ICL-corpus and WORDS-corpus. Both of them adopt the Text REtrieval Conference (TREC) format [9], i.e. every document is marked up by <DOC></DOC> tags and has a unique document identifier which is marked up by <DOCNO></DOCNO> tags.

The ICL-corpus consists of 1,000 documents which came from Indonesian Choral Lovers Yahoo! Groups, a mailing list of Indonesian choral community, hence the body text is the body of email. During annotation process, each document of ICL-corpus has been assigned a topic, or more, by choral experts and concurrently they were expected to determine words that highly related with the topics given [15]. We then treated those keywords as the body text of document in WORDS-corpus. Therefore, both corpora are basically correlated in the sense that WORDS-corpus contains keywords defined by human experts for the given topic(s) of each document in ICL-corpus. Hence, the WORDS-corpus also consists of 1,000 documents and the identifier of WORDS-corpus' document is in accordance with identifier of ICL-corpus' document.

We take an assumption that each topic given by the human experts in annotation process is a concept, therefore we consider the keywords determined by them as the term variants that semantically related with particular concept. These keywords are in WORDS-corpus, hence the WORDS-corpus contains important terms of particular concept selected by human expert. In automatic process of the system, these terms should be selected, therefore WORDS-corpus become the ground truth of this study.

Topic assignment yielded 127 topics which many of them has few document frequency; 81.10 % have document frequency less than 10 and 32.28 % of them have document frequency 1.

4 Methodology

A thesaurus is a type of lightweight ontology which provides additional relationships, and does not provide an explicit hierarchy [6]. By definition, a thesaurus could be represented technically in the form of a term-by-term matrix. In this study, we considered the *tolerance matrix* generated using TRSM as the representation of the intended thesaurus, which is technically a term-by-term matrix and contains tolerance classes of all index terms.

Figure 1 shows the main phases of the study, which were performed twice: with stemming task and without stemming task.

4.1 Extraction Phase

The main objective of extraction phase was preprocessing both corpora. A version of Indonesian stemmer, called *CS stemmer*, was employed in stemming task. In [1], it was introduced as a new confix-stripping approach for automatic Indonesian stemming and was showed as the most accurate stemmer among other

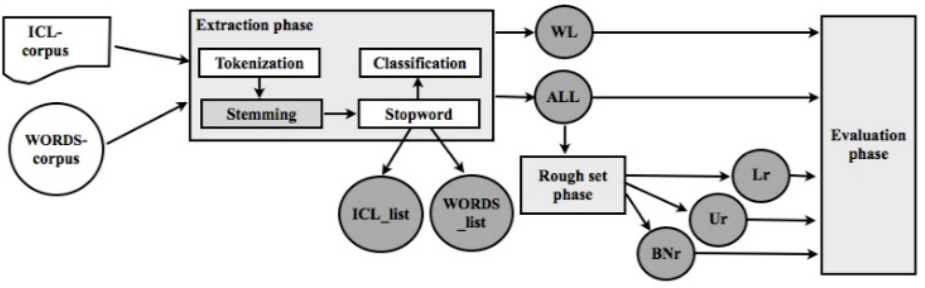


Fig. 1. Main phases of the study: extraction phase, rough set phase, and evaluation phase. A rectangle represents a phase while a circle represent a result.

automated Indonesian stemmer. For stopword task, Vega's stopword [14] was applied as in [2] the use of the stopword gave highest precision and recall.

Documents were tokenized based on character other than alphabetic. The resulted tokens were stemmed using the CS stemmer and then compared to the Vega's stopword. It yielded list of unique terms and its frequency. There were 9,458 unique terms extracted from ICL-corpus and 3,390 unique terms extracted from WORDS-corpus; called *ICL_list* and *WORDS_list* respectively. When it was run without stemming process, we identified 12,363 unique terms in ICL_list and 4,281 unique terms in WORDS_list.

Both corpora were classified based on 127 topics yielded in preliminary process. Taking the assumption that keywords determined by human experts are the term variants of a concept then aggregation of all terms appeared in each class were taken as the terms of representative vector of each class. The resulted classes of ICL-corpus was called *ALL* while the resulted classes of WORDS-corpus was called *WL*. The frequency matrix of topic-term needed in rough set phase was created based on these classes.

4.2 Rough Set Phase

This phase was conducted in order to generate the lower set *Lr*, upper set *Ur*, and boundary set *BNr* of each class; *RS* refers to all three sets. These sets were possible to be created using (1), (2), and (3) when tolerance matrix was ready.

The tolerance matrix was created based on algorithm explained in [8]. It needed topic-term frequency matrix as the input, then the occurrence binary matrix *OC matrix*, co-occurrence matrix *COC matrix*, and tolerance binary matrix *TOL matrix* were generated in sequence manner by employing (4), (5), and (6) respectively. Note that $tf_{i,j}$ denotes the frequency of term j in topic i and θ is the co-occurrence threshold of terms.

$$oc_{i,j} = 1 \quad \Leftrightarrow \quad tf_{i,j} > 0 \ . \tag{4}$$

$$coc_{x,y} = \text{card}(OC^x \text{ AND } OC^y) \ . \tag{5}$$

$$tol_{x,y} = 1 \quad \Leftrightarrow \quad coc_{x,y} \geq \theta \ . \tag{6}$$

4.3 Evaluation Phase

In this phase, all resulted sets were compared across the other, i.e. ICL_list vs. WORDS_list, ALL vs. WL, ALL vs. RS, WL vs. RS, and between RS (Lr vs. BNr, Ur vs. Lr, and Ur vs. BNr). The objective is to get the amount of terms appeared in both compared sets. These comparisons were conducted for co-occurrence threshold θ between 1 to 75. From each comparison at particular θ value, we got 127 values which were the value of each class. The average value was then computed for each θ value as well as for all θ value.

Recall and precision are measures commonly used in information retrieval field to evaluate the system performance. Recall R is the fraction of relevant documents that are retrieved while precision P is the fraction of retrieved documents that are relevant [7]. Suppose Rel denotes relevant documents and Ret denotes retrieved documents, then recall R and precision P are defined as follow

$$R = \frac{|Rel \bigcap Ret|}{|Rel|} \qquad P = \frac{|Rel \bigcap Ret|}{|Ret|} \ . \tag{7}$$

In this study, both measures were used for the *terms* rather than *documents*. That is to say, by considering WL as the ground truth, then recall R is the fraction of relevant terms that are retrieved while precision P is the fraction of retrieved terms that are relevant. Based on the definition, better recall value is preferred than better precision value because better recall value will ensure the availability of important terms in the set.

5 Analysis

With regard to the process of developing WORDS_list, the fact that ICL_list could cover almost all WORDS_list terms was not surprising. It was interesting though that there were some terms of WORDS_list did not appear in ICL_list; 17 terms yielded by the process without stemming task and 11 terms yielded by the process with stemming task. By examining those terms, we found that the *CS stemmer* could only handle the formal terms (6 terms) and left the informal terms (5 terms) as well as the foreign term (1 term); the other terms caused by typographical error (5 terms) in ICL_corpus.

Despite the fact that CS stemmer succeeded in reducing the number of terms of ICL_list (23.50%) as well as of WORDS_list (20.81%), it reduced the average of recall in each class of ALL about 0.64% from 97.39%. We noticed that the average of precision in each class of ALL increased about 0.25%, however the values themselves were very small (14.56% for process without stemming task and 14.81% for process with stemming task). From these, we could say that the ICL_list was still too noisy of containing many unimportant terms in describing particular topic.

5.1 ALL vs. RS

Table 1 shows the average values of comparison process between *ALL vs. RS* and *WL vs. RS* in percentage. The values of ALL-Ur for process with and without

Table 1. Average of Co-occurrence Terms Between Sets

	With Stemming			Without Stemming		
	Ur (%)	Lr (%)	BNr (%)	Ur (%)	Lr (%)	BNr (%)
(1)	(2)	(3)	(4)	(5)	(6)	(7)
ALL	100.00	5.00	95.00	100.00	4.43	95.57
WL_{Recall}	97.64	5.55	92.08	97.55	4.64	92.91
$WL_{Precision}$	13.77	27.49	13.50	14.13	26.30	13.75

stemming task, which are 100%, made us confident that the TRSM model has been employed correctly.

The low values of ALL-Lr (5% and 4.43%) and the high values of ALL-BNr (95% and 95.57%) compared with the low values of WL_{Recall}-Lr (5.55% and 4.64%) and the high values of WL_{Recall}-BNr (92.08% and 92.91%) indicate that rough sets theory seemed to work in accordance with the natural way of human thinking. From the values of WL_{Recall}, we could learn that it was possibly the case happened during topic assignment, that only limited number of terms could be considered precisely belong to a particular topic while numerous of others could not, e.g. were in uncertain condition. It was supported by the fact that many times the human experts seemed to encounter difficulty in determining keywords during annotation process. We came into this from the data that rather than listing the keywords, they chose sentences on the text or even made their own sentences. By doing this, they did not define specific terms as the highly related terms with particular topic but mentioning many other terms in the form of sentences instead. From this, we can say that the rough set theory is able to model the natural way of topic assignment conducted by human.

From Table 1, we can see that all values in column 3 are higher than all values in column 6 while all values in column 4 are lower than all values in column 7. Hence, it seems that employing stemming task could retrieve more terms considered as the "core" terms of a concept and at the same time reduce the number of uncertain terms retrieved.

5.2 WL vs. RS

Table 1 shows us that value of WL_{Recall}-Ur of process with stemming is higher than the process without stemming. It supports our confidence so far that stemming task with CS stemmer would bring more benefit in this framework of study.

Despite the fact that better recall is preferred than better precision, as we explained in 4.3, we noticed that the values of $WL_{Precision}$-Ur are small (13.77% and 14.13%). With regard to (7), they were calculated using equation $P = \frac{|WL \bigcap Ur|}{|Ur|}$. Based on the equation, we can expect to improve the precision value by doing one, or both, of these: (1) increasing the co-occurence terms of WL and Ur or (2) decreasing the total number of Ur. Suppose we have a constant number of Ur (after setting up the θ at a certain value), then what we should

do to improve the precision is increasing the number of co-occurence terms, i.e. increasing the availability of relevant terms in Ur.

It has been explained in section 4.2 that the topic-term frequency matrix was used as the input of generating the tolerance class. It means, the weighting scheme was solely based on the term frequency of occurrence in particular topic. By this fact, the precision value is possible to be enhanced by improving the weighting scheme.

5.3 Tolerance Value

From ALL-Ur comparison with stemming, we also found that there was indication that Ur set enriched ALL set; it was based on the average value of co-occurrence terms over Ur that was 70.79% for θ value 1 to 75. In fact, the average value was started from 4.02% for $\theta = 1$ and getting higher up to 99.33% for $\theta = 75$. Note that the smaller the average value means the possibility of Ur set enrichs ALL set is higher. With regard to (6), it is reasonable that increasing the θ value will reduce the number of total terms in Ur set and increase the average value of co-occurrence terms between ALL-Ur over Ur. The important point of this is the possibility of enriching a concept getting lower by increasing the θ value.

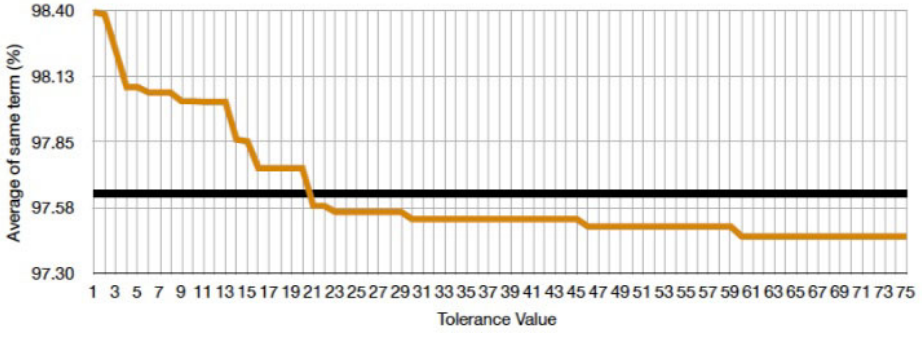


Fig. 2. The WL_{Recall}-Ur comparison

Figure 2 is the graph of co-occurrence terms between WL set and Ur set over WL for θ value 1 to 75. It is clear that after dramatic changes the graph starts to stable at tolerance value 21. Looking at the average number of terms in Ur set at $\theta = 21$ was also interesting. It is 733.79 terms, which means reducing 92.24% of the average number of terms in Ur set at $\theta = 0$ which is 9.458 terms. From Fig. 2, we can also see that the average number of co-occurrence terms at $\theta = 21$ is 97.58%, which is high. By this manual inspection, we are confident to propose $\theta \geq 21$ to be used in similar framework of study. However, automatically setting the tolerance value is suggested, especially while considering the nature of mailing list, i.e. growing over the time.

5.4 ICL_list vs. Lexicon

Lexicon is vocabulary of terms [7]. The lexicon used by CS stemmer in this study consists of 29,337 Indonesian base words. Comparison between ICL_list and Lexicon showed that there was 3,321 co-occurrence terms. In other words, 64.89% of ICL_list was different from Lexicon. Out of 6,137 terms, we analyzed the top 3,000 terms with respect to the document frequency.

We identified that the biggest problem (37.3% of terms) was caused by foreign language; most of them was English. Next problems were the colloquial terms which was 26.1% of terms and proper nouns which was 22.73% of terms. Combination of foreign and Indonesian terms, e.g. *workshopnya*, was considered as colloquial terms. We also found that the CS stemmer should be improved as there were 19 formal terms left unstemmed in ICL_list. Finally, we suggested 5 terms to be added into Lexicon and 8 terms into stopword-list.

6 Conclusion

This paper was thought as the fundamental basis that generating an automatic thesaurus using rough sets theory is a promising way. There was indication that it could enrich a concept and proved to be able to cover the important terms that should be retrieved by automatically process of system, even though foreign languages, colloquial terms and proper nouns were identified as big problems in main corpus. We noticed that CS stemmer as a version of Indonesian stemming algorithm was able to reduce the total number of index terms being processed and improved the recall of Ur as it was expected, however it should be upgraded. In this paper, we also proposed θ value ≥ 21 for similar framework of study, as well as suggesting some terms to be added into Lexicon and stopword-list.

There is much work to do related with this study, such as (1) to improve the weighting scheme hence it is not only based on term frequency of occurrence, (2) to upgrade the CS stemmer to handle the formal terms better, (3) to find a way in dealing with the foreign terms and colloquial terms, and (4) to set the θ value automatically, particularly by considering the nature of mailing list.

Acknowledgments. This work is partially supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information", specific Grant Agreement Number-2008-4950/001-001-MUN-EWC from European Union Erasmus Mundus "External Cooperation Window" EMMA, and grants from Ministry of Science and Higher Education of the Republic of Poland (N N516 368334 and N N516 077837). We also thank Faculty of Computer Science, University of Indonesia, for the permission of using the CS stemmer.

References

1. Adriani, M., Asian, J., Nazief, B., Tahaghogi, S.M.M., Williams, H.E.: Stemming Indonesian: A Confix-Stripping Approach. ACM Transactions on Asian Language Information Processing 6(4), 1–33 (2007), Article 13
2. Asian, J.: Effective Techniques for Indonesian Text Retrieval. Doctor of Philosophy Thesis. School of Computer Science and Information Technology. RMIT University (2007)
3. Gaoxiang, Y., Heling, H., Zhengding, L., Ruixuan, L.: A Novel Web Query Automatic Expansion Based on Rough Set. Wuhan University Journal of Natural Sciences 11(5), 1167–1171 (2006)
4. Kawasaki, S., Nguyen, N.B., Ho, T.B.: Hierarchical Document Clustering Based on Tolerance Rough Set Model. In: 4th European Conference on Principles of Data Mining and Knowledge Discovery, pp. 458–463. Springer, London (2000)
5. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough Sets: A Tutorial. In: Rough Fuzzy Hybridization: A New Trend in Decision-Making, pp. 3–98. Springer, Singapore (1998)
6. Lassila, O., McGuinness, D.: The Role of Frame-Based Representation on the Semantic Web. Technical Report KSL-01-02, Knowledge System Laboratory, Standford University (2001)
7. Manning, C.D., Raghavan, P., Schutze, H.: An Introduction to Information Retrieval. Cambridge University Press, England (2009)
8. Nguyen, H.S., Ho, T.B.: Rough Document Clustering and the Internet. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 987–1003. John Wiley & Sons Ltd., Chichester (2008)
9. National Institute of Standards and Technology, http://www.nist.gov/srd/nistsd23.cfm
10. 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)
11. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Science 11(5), 341–356 (1982)
12. 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)
13. Skowron, A., Stepaniuk, J.: Tolerance Approximation Spaces. Fundam. Inf. 27(2-3), 245–253 (1996)
14. Vega, V.B.: Information Retrieval for the Indonesian Language. Master thesis. National University of Singapore (2001) (unpublished)
15. Virginia, G., Nguyen, H.S.: Automatic Ontology Constructor for Indonesian Language. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 440–443. IEEE Press, Los Alamitos (2010)
16. Virginia, G., Nguyen, H.S.: Investigating the Potential of Rough Sets Theory in Automatic Thesaurus Construction. In: 2011 International Conference on Data Engineering and Internet Technology, pp. 882–885. IEEE, Los Alamitos (2011)

Report of the ISMIS 2011 Contest: Music Information Retrieval

Bozena Kostek[1], Adam Kupryjanow[1], Pawel Zwan[1], Wenxin Jiang[2], Zbigniew W. Raś[3,4], Marcin Wojnarski[5], and Joanna Swietlicka[5]

[1] Multimedia Systems Department, Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, PL
{bozenka,adamq,zwan}@sound.eti.pg.gda.pl
[2] Fred Hutchinson Cancer Research Center, Seattle, WA 98109, USA
wjiang2@fhcrc.org
[3] Univ. of North Carolina, Dept. of Computer Science, Charlotte, NC 28223, USA
[4] Warsaw Univ. of Technology, Institute of Comp. Science, 00-665 Warsaw, Poland
ras@uncc.edu
[5] TunedIT Solutions, Zwirki i Wigury 93/3049, 02-089 Warszawa, Poland
{marcin.wojnarski,j.swietlicka}@tunedit.org

Abstract. This report presents an overview of the data mining contest organized in conjunction with the 19th International Symposium on Methodologies for Intelligent Systems (ISMIS 2011), in days between Jan 10 and Mar 21, 2011, on TunedIT competition platform. The contest consisted of two independent tasks, both related to music information retrieval: recognition of music genres and recognition of instruments, for a given music sample represented by a number of pre-extracted features. In this report, we describe aim of the contest, tasks formulation, procedures of data generation and parametrization, as well as final results of the competition.

Keywords: Automatic genre classification, instrument recognition, music parametrization, query systems, intelligent decision systems.

1 Introduction

Internet services expose nowadays vast amounts of multimedia data for exchange and browsing, the most notable example being YouTube. These digital databases cannot be easily searched through, because automatic understanding and indexing of multimedia content is still too difficult for computers – take, for example, the diversity of musical trends and genres, uncommon instruments or the variety of performers and their compositions present in typical multimedia databases. In ISMIS 2011 Contest, we invited all researchers and students interested in sound recognition and related areas (signal processing, data mining, machine learning) to design algorithms for two important and challenging problems of Music Information Retrieval: recognition of music genres (jazz, rock, pop, ...) and recognition of instruments playing together in a given music sample.

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 715–724, 2011.

The contest comprised 2 tracks:

- Music Genres – Automatic recognition of music genre (jazz, rock, pop, ...) from a short sample. The dataset and feature vectors were prepared by the Multimedia Systems Department of the Gdansk University of Technology.
- Music Instruments – Automatic recognition of instruments playing together in a given sample. Prepared by Wenxin Jiang and Zbigniew Raś.

The tasks were independent. Contestants could have participated in both of them or in a selected one. The challenge was organized at TunedIT Challenges[1] platform as an on-line *interactive* competition: participants were submitting solutions many times, for the whole duration of the challenge; solutions were immediately automatically evaluated and results were published on the Leaderboard, to allow comparison with other participants and introduction of further improvements. TunedIT Challenges is an open platform that can be freely used by everyone for scientific and didactic purposes [23].

Contestants were given descriptions of both tasks along with vectors of parameters – features extracted from raw sound samples. For each track, the full set of vectors was divided into two subsets: (1) *training set*, disclosed publicly to participants for classifier building; (2) *test set*, disclosed without decisions, which had to be predicted by contestants – the predictions were subsequently compared with ground truth kept on the server and scores were calculated. Test set was further divided into preliminary (35%) and final (65%) parts: the former being used for calculating results shown on the Leaderboard during contest; the latter one used for final unbiased scoring and picking up the winner.

The competition attracted very large interest among Data Mining and Music Information Retrieval community: 292 teams with 357 members had registered, 150 of them actively participated, submitting over 12.000 solutions in total, largely outperforming baseline methods. The winners are:

- Music Genres: **Amanda C. Schierz and Marcin Budka**, Bournemouth University, UK. Achieved 59% lower error rate than baseline algorithm.
- Music Instruments: **Eleftherios Spyromitros Xioufis**, Aristotle University of Thessaloniki, Greece. Achieved 63% lower error than baseline.

The winning teams were awarded prizes of 1000 USD each. See also the contest web page: `http://tunedit.org/challenge/music-retrieval`.

2 Task 1: Music Genres Recognition

2.1 Database and Parametrization Methods

The automatic recognition of music genres is described by many researchers [1,3,4,6,7,11,15,16,17,18,20,21], who point out that the key issue of this process is a proper parametrization. Parametrization has so far experienced extensive

[1] http://tunedit.org

development [5,8,9,10,11,12,24], however, there are still some important areas of Music Information Retrieval, such as for example music genre classification, that is researching this aspect. The parameters that are most often used are MPEG-7 descriptors [6], mel cepstral coefficients [19] and bit histograms [18]. The classification is typically based on the analysis of a short (3–10 sec.) excerpts of a musical piece which are parameterized and later classified.

A database of 60 music musicians/performers was prepared for the competition. The material is divided into six categories: classical music (class No. 1), jazz (class No. 2), blues (class No. 3), rock (class No. 4), heavy metal (class No. 5) and pop (class No. 6). For each of the performers 15–20 music pieces were collected. The total number of music pieces available for each of music genres is presented in Tab. 1. An effort was put to select representative music pieces for each of music genres.

All music pieces are partitioned into 20 segments and parameterized. The descriptors used in parametrization also those formulated within the MPEG-7 standard, are only listed here since they have already been thoroughly reviewed and explained in many research studies.

Table 1. The number of music pieces extracted for a given genre

Genre	Number of music pieces
Classical music	320
Jazz music	362
Blues	216
Rock music	124
Heavy metal music	104
Pop music	184

For each of music pieces 25-second-long excerpts were extracted and parameterized. The excerpts are evenly distributed in time. Each of these excerpts was parameterized and the resulting feature vector contains 171 descriptors. 127 of the features used are MPEG-7 descriptors, 20 are MFCC and additional 24 are time-related 'dedicated' parameters. Those 'dedicated' parameters are original authors' contribution to the parametrization methods. Since MPEG-7 features and mel-frequency cepstral-coefficients are widely presented in a rich literature related to this subject there they will only be listed. Dedicated parameters will be described in the next Section.

The parameters utilized are listed below:

a) parameter 1: Temporal Centroid,
b) parameter 2: Spectral Centroid average value,
c) parameter 3: Spectral Centroid variance,
d) parameters 4–37: Audio Spectrum Envelope (ASE) average values in 34 frequency bands,
e) parameter 38: ASE average value (averaged for all frequency bands),

f) parameters 39–72: ASE variance values in 34 frequency bands,
g) parameter 73: averaged ASE variance parameters,
h) parameters 74, 75: Audio Spectrum Centroid – average and variance values,
i) parameters 76, 77:Audio Spectrum Spread – average and variance values,
j) parameters 78–101:Spectral Flatness Measure (SFM) average values for 24 frequency bands,
k) parameter 102: SFM average value (averaged for all frequency bands),
l) parameters 103–126: Spectral Flatness Measure (SFM) variance values for 24 frequency bands,
m) parameter 127: Averaged SFM variance parameters,
n) parameters 128–147: 20 first mel-frequency cepstral coefficients (mean values),
o) parameters 168–191: Dedicated parameters in time domain based on the analysis of the distribution of the envelope in relation to the rms value.

The dedicated parameters are related to the time domain. They are based on the analysis of the distribution of sound sample values in relation to the root mean square values of the signal (rms). For this purpose three reference levels were defined: r_1, r_2, r_3 – equal to namely 1, 2, 3 rms values of the samples in the analyzed signal frame.

The first three parameters are related to the number of samples that are exceeding the levels: r_1, r_2 and r_3.

$$p_n = \frac{count(samples\_exceeding\_r_n)}{length(x(k))} \tag{1}$$

where $n = 1, 2, 3$ and $x(k)$ is the signal frame analyzed.

The initial analysis of the values of parameters p_n showed a difficulty, because the rms level in the excerpts analyzed sometimes significantly varies within the analyzed frame. In order to cope with this problem another approach was introduced. Each 5-second frame was divided into 10 smaller segments. In each of these segments parameters p_n (Eq. 1) were calculated. As a result a sequence P_n was obtained:

$$P_n = \{p_n^1, p_n^2, p_n^3, ..., p_n^{10}\} \tag{2}$$

where p_n^k, $k = 1 \ldots 10$ and $n = 1, 2, 3$ as defined in Eq. 1.

In this way, 6 new features were defined on the basis of sequences P_n. New features were defined as mean (q_n) and variance (v_n) values of P_n, $n = 1, 2, 3$. The index n is related to the different reference values of r_1, r_2 and r_3.

$$q_n = \frac{\sum_{k=1}^{10} p_n^k}{10} \tag{3}$$

$$v_n = var(P_n) \tag{4}$$

In order to supplement the feature description, additional three parameters were defined. They are calculated as the 'peak to rms' ratio, but in 3 different ways described below:

- parameter k_1 calculated for the 5-second frame,
- parameter k_2 calculated as the mean value of the ratio calculated in 10 sub-frames,
- parameter k_3 calculated as the variance value of the ratio calculated in 10 sub-frames.

The last group of dedicated parameters is related to the observation of the rate of the threshold value crossing. This solution may be compared to the more general idea based on classical zero crossing rate parametrization (ZCR). The ZCR parametrization is widely used in many fields related to automatic recognition of sound. The extension of this approach is a definition of a threshold crossing rate value (TCR) calculated analogically as ZCR, but by counting the number of signal crossings in relation not only to zero, but also to the r_1, r_2 and r_3 values. These values (similarly as in the case of the other previously presented parameters) are defined in 3 different ways: for the entire 5-second frame and as the mean and variance values of the TCR calculated for 10 sub-frames. This gives 12 additional parameters to the feature set.

The entire set of dedicated parameters consists of 24 parameters that supplement 147 parameters calculated based on MPEG-7 and mel-cepstral parametrization.

2.2 Database Verification

Before the publication of the database for the purpose of the contest the data collected needed to be tested. The set of feature vectors was divided into two parts: training and test set, with a proportion of 50:50 in such a way that excerpts extracted from one music piece could belong either to the training or to the test set (in order to test effectiveness of the system for recognizing music pieces that were not used during the training).

Next, three classifiers, namely: SVM (Support Vector Machine), J48 tree and Random Forest were trained on the basis of these data. Those classifiers were verified on a training data. In the case of SVM classifiers the C-SVC SVM with the Radial Basis kernel function was used. Settings of the algorithm as well as parameters of the kernel were chosen experimentally. The cost parameter was set to 62.5 and gamma for the kernel to 0.5. The tolerance of the termination criterion (epsilon) was equal to 10e-3. In addition, linear kernel was tested to check if the parameters could be separated linearly. Since no classification accuracy improvement has been noticed, thus the radial basis was used. For this classifier the libSVM library [2] linked to the WEKA environment [22] was employed.

Random Forest and Decision Tree classifiers were tested in the WEKA system. The Random Forest model was created with the unlimited depth of trees and the number of trees was equal to 20. For the J48 the confidence factor used for the pruning was set to 0.25 and the minimum number of instances per leaf was equal to 2.

In Tab. 2 the average efficiency obtained for these three classifiers is presented. The best results were obtained for the SVM classifier due to its ability to the non-linear separation of classes.

Table 2. Comparison of the average efficiency for 3 classifiers (first experiment)

Classifier	**Average efficiency** [%]
Support Vector Machine	**90.87**
J48	77.40
Random Forest	84.72

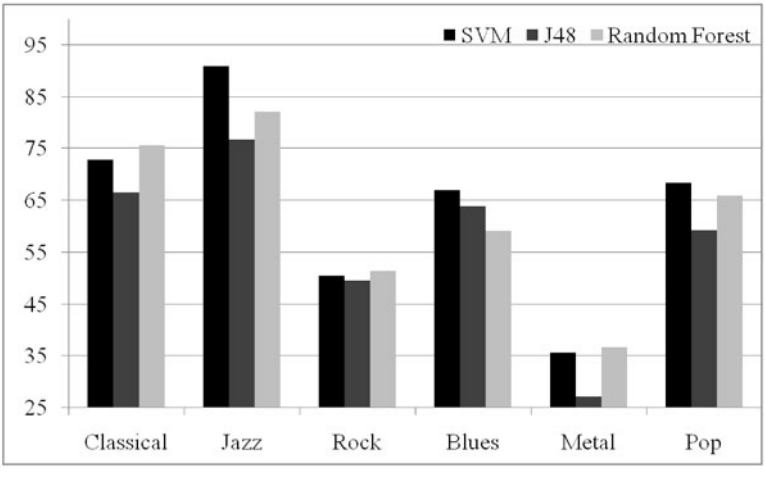


Fig. 1. Classification efficiency of music genre using three classifiers (second experiment)

Table 3. Comparison of the average efficiency for 3 classifiers (second experiment)

Classifier	**Average efficiency** [%]
Support Vector Machine	**70.0**
J48	62.1
Random Forest	67.3

The second experiment was related to different division of data: excerpts extracted from one music artist/performer could belong either to the training or to the test set. In this case smaller accuracy was expected than in the first scenario, since testing was performed on excerpts that were not earlier used in the training phase. Results of this experiment are presented in Fig. 1 and Tab. 3. The same classifier settings as earlier described were used in this experiment.

Similarly to the previous experiment, the best results were obtained by the SMV classifier. As was expected the results had smaller efficiency. In addition, the confusion matrix for the SVM classifier is presented in Tab. 4.

Table 4. Confusion matrix obtained in the second experiment

[%]	Classical	Jazz	Rock	Blues	H. Metal	Pop
Classical	**72**	28	0	0	0	0
Jazz	8	**91**	0.5	0.5	0	0
Rock	0.5	21.5	**50.5**	7	14	7
Blues	3	3	1	**67**	10.5	15
H. Metal	0	2	42	6	**35**	14
Pop	0.5	2.5	11	13	5	**68.5**

The diagonal of the matrix shows the efficiency of automatic recognition for each of the classes. However the main misclassifications are additionally marked (underlined). The 'classical music' class was mainly misclassified as 'jazz' genre (28%). It is worth mentioning that any classical excerpts were classified as rock, blues, heavy metal, or pop genres. Similarly, nearly all misclassification for jazz excerpts were related to the classical music. Other genre misclassification groups are {Jazz, Rock, Heavy Metal} {Heavy Metal, Rock}, so it can be concluded that those misclassifications are related to the similarity between music genres.

The Organizers of the competition have decided to use the data divided as in the second experiment – excerpts extracted from one music performer could belong either to the training or to the test set, thus one could expect results close or better to the ones obtained in our preliminary experiment.

3 Task 2: Music Instruments Recognition

In recent years, rapid advances in digital music creation, collection and storage technology have enabled organizations to accumulate vast amounts of musical audio data, which are manually labeled with some description information, such as title, author, company, and so on. However, in most cases those labels do not have description on timbre or other perceptual properties and are insufficient for content-based searching.

Timbre is the quality of sound that distinguishes different musical instruments playing the same note with the identical pitch and loudness. Recognition and separation of sounds played by various instruments is very useful in labelling audio files with semantic information. However timbre recognition, one of the main subtasks of Music Information Retrieval, has proven to be extremely challenging especially in multi-timbre sounds, where multiple instruments are playing at the same time.

It is fairly easy to automatically recognize single instruments. However, when minimum two instruments are playing at the same time, the task becomes much more difficult. In this contest, the goal is to build a model based on data collected for both single instruments and examples of mixtures in order to recognize pairs of instruments.

3.1 Data

The data are taken from the musical sounds of MUMS (McGill Univ. Master Samples) and samples recorded in the KDD Lab at UNC Charlotte. The original audio sounds have a format of a large volume of unstructured sequential values, which are not suitable for traditional data mining algorithms. The process of feature extraction is usually performed to build a table representation from the temporal or spectral space of the signal. This will reduce the raw data into a smaller and simplified representation while preserving the important information for timbre estimation. Sets of acoustical features have been successfully developed for timbre estimation in monophonic sounds where single instruments are playing [14].

Training data consists of two datasets: large one containing data for single instruments and much smaller one containing mixtures of pairs of different instruments. 26 music instruments are involved : electric guitar, bassoon, oboe, b-flat clarinet, marimba, c trumpet, e-flat clarinet, tenor trombone, French horn, flute, viola, violin, English horn, vibraphone, accordion, electric bass, cello, tenor saxophone, b-flat trumpet, bass flute, double bass, alto flute, piano, Bach trumpet, tuba, and bass clarinet. Both sets have the same attributes (the single instruments set contains some additional information, though).

The test set contains only data for mixtures, which are the features extracted from 52 music recording pieces synthesized by Sound Forge sound editor, where each piece was played by two different music instruments. The pairs of instruments that are playing in each piece are to be predicted. Note that test and training sets contain different pairs of instruments (i.e. the pairs from the training set do not occur in the test set). Moreover, not all instruments from the training data must also occur in the test part. There also may be some instruments from the test set that only appear in the single instruments part of the training set.

The following attributes are used to represent the data:

- BandsCoef1-33, bandsCoefSum - Flatness coefficients
- MFCC1-13 - MFCC coefficients
- HamoPk1-28 - Harmonic peaks
- Prj1-33, prjmin, prjmax, prjsum, prjdis, prjstd - Spectrum projection coefficients
- SpecCentroid, specSpread, energy, log spectral centroid, log spectral spread, flux, rolloff, zerocrossing - The other
- acoustic spectral features
- LogAttackTime, temporalCentroid - Temporal features

These attributes are the acoustic features either in the frequency domain or in the time domain. Some of the features are extracted according to the MPEG-7 standard [14](such as spectral flatness coefficients and spectral centroid). Other non-MPEG7 features are also extracted such as MFCC (Mel frequency cepstral coefficients) which describes the spectrum according to the human perception system in the mel scale [13]. Additionally, the single instruments set contains:

- Frameid - Each frame is 40ms long signal
- Note - Pitch information
- Playmethod - One schema of musical instrument classification according to the way they are played
- Class1,class2 - Another schema of musical instrument classification according to Hornbostel-Sachs

Both sets contain also the actual labels, i.e. the instruments. For the mixture data, there are two instruments, thus - two labels.

3.2 Solutions and Evaluation

The goal of the contest is to select the best features provided in the training set to build the appropriate classifier that yields the high confidence of the instrument classification for the test set.

Solution should be a text file containing one pair of labels per line. The names of the instruments in each line can be given in any order and should be separated by a comma. Different number of labels in a line will result in an error. Labels are not case sensitive. The evaluation metric is modified accuracy:

- If no recognized instrument matches the actual ones, 0.0 score is assigned
- If only one instrument is correctly recognized, 0.5 is assigned
- If both instruments match the target ones, 1.0 is assigned The final score is equal to arithmetic mean of individual scores.

Acknowledgements

This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information". It is supported by the grants N N516 368334 and N N516 077837 from the Ministry of Science and Higher Education of the Republic of Poland. Also, this work is supported by the National Science Foundation under Grant Number IIS-0968647.

References

1. Aucouturier, J.-J., Pachet, F.: Representing musical genre: A state of art. Journal of New Music Research 32(1), 83–93 (2003)
2. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
3. Fingerhut, M.: Music Information Retrieval, or how to search for (and maybe find) music and do away with incipits. In: IAML-IASA Congress, Oslo, August 8-13 (2004)
4. Foote, J.: Content-based retrieval of music and audio. Multimed. Storage Archiv. Syst. II, 138–147 (1997)

5. Herrera, P., Amatriain, X., Batlle, E., Serra, X.: Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In: International Symposium on Music Information Retrieval, ISMIR (2000)
6. Hyoung-Gook, K., Moreau, N., Sikora, T.: MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval. Wiley & Sons, Chichester (2005)
7. The International Society for Music Information Retrieval/Intern. Conference on Music Information Retrieval website, http://www.ismir.net/
8. Kostek, B., Czyzewski, A.: Representing Musical Instrument Sounds for their Automatic Classification. J. Audio Eng. Soc. 49, 768–785 (2001)
9. Kostek, B.: Soft Computing in Acoustics, Applications of Neural Networks. In: Fuzzy Logic and Rough Sets to Musical Acoustics. Studies in Fuzziness and Soft Computing. Physica Verlag, Heildelberg (1999)
10. Kostek, B.: Perception-Based Data Processing in Acoustics. In: Applications to Music Information Retrieval and Psychophysiology of Hearing. Series on Cognitive Technologies, Springer, Heidelberg (2005)
11. Kostek, B., Kania, L.: Music information analysis and retrieval techniques. Archives of Acoustics 33(4), 483–496 (2008)
12. Lindsay, A., Herre, J.: MPEG-7 and MPEG-7 Audio – An Overview, vol. 49(7/8), pp. 589–594 (2001)
13. Logan, B.: Mel Frequency Cepstral Coefficients for Music Modeling. In: Proceedings of the First Int. Symp. on Music Information Retrieval, MUSIC IR 2000 (2000)
14. O/IEC JTC1/SC29/WG11. MPEG-7 Overview (2004), http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
15. Pachet, F., Cazaly, D.: A classification of musical genre. In: Proc. RIAO Content-Based Multimedia Information Access Conf., p. 2000 (2003)
16. Panagakis, I., Benetos, E., Kotropoulos, C.: Music Genre Classification: A Multilinear Approach. In: Proc. Int. Symp. Music Information Retrieval, ISMIR 2008 (2008)
17. Pye, D.: Content-based methods for the management of digital music. In: Proc. Int. Conf. Acoustics, Speech, Signal Processing, ICASSP (2000)
18. Scheirer, E.D.: Tempo and beat analysis of acoustic musical signals. J. Acoust. Soc. Am. 103(1) (January 1998)
19. Tyagi, V., Wellekens, C.: On desensitizing the Mel-Cepstrum to spurious spectral components for Robust Speech Recognition. In: Proc. ICASSP 2005, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 529–532 (2005)
20. Tzanetakis, G., Essl, G., Cook, P.: Automatic musical genre classification of audio signals. In: Proc. Int. Symp. Music Information Retrieval, ISMIR (2001)
21. Tzanetakis, G., Cook, P.: Musical genre classification of audio signal. IEEE Transactions on Speech and Audio Processing 10(3), 293–302 (2002)
22. WEKA, http://www.cs.waikato.ac.nz/ml/weka/
23. Wojnarski, M., Stawicki, S., Wojnarowski, P.: TunedIT.org: System for automated evaluation of algorithms in repeatable experiments. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 20–29. Springer, Heidelberg (2010)
24. Zwan, P., Kostek, B.: System for Automatic Singing Voice Recognition. J. Audio Eng. Soc. 56(9), 710–723 (2008)

High–Performance Music Information Retrieval System for Song Genre Classification

Amanda Schierz* and Marcin Budka

Smart Technology Research Centre, School of Design, Engineering and Computing, Bournemouth University, Poole House, Talbot Campus, Fern Barrow, Poole BH12 5BB, United Kingdom
{aschierz,mbudka}@bournemouth.ac.uk

Abstract. With the large amounts of multimedia data produced, recorded and made available every day, there is a clear need for well–performing automatic indexing and search methods. This paper describes a music genre classification system, which was a winning solution in the Music Information Retrieval ISMIS 2011 contest. The system consisted of a powerful ensemble classifier using the Error Correcting Output Coding coupled with an original, multi–resolution clustering and iterative relabelling scheme. The two approaches used together outperformed other competing solutions by a large margin, reaching the final accuracy close to 88%.

Keywords: error correcting output codes, multi–resolution clustering, music information retrieval, semi–supervised learning.

1 Introduction

Internet services expose vast amounts of multimedia data. These digital libraries cannot be easily searched through, because automatic understanding and indexing of multimedia content is still too difficult for computers.

In order to stimulate the research in this area, the ISMIS 2011 Contest: Music Information Retrieval[1] associated with the 19^{th} International Symposium on Methodologies for Intelligent Systems[2], has been organized. The task was to recognize different properties of provided music samples, based on extracted sound features. The contest consisted of two independent tracks: music genre recognition and music instrument recognition. This paper describes an advanced data mining algorithm, which has proven to be a winning solution in the music genre recognition track.

2 Dataset Properties

The dataset used in the music genre track of the competition was a database of 60 music performers. The material has been divided into six categories: classical

* Corresponding author.

[1] http://tunedit.org/challenge/music-retrieval

[2] http://ismis2011.ii.pw.edu.pl/

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 725–733, 2011.

music, jazz, blues, pop, rock and heavy metal. For each of the performers between 15 to 20 music pieces have been collected. All music pieces are partitioned into 20 segments and parameterized.

The feature vector consisted of 191[3] parameters. The first 127 parameters were based on the MPEG–7 standard. The remaining ones were cepstral coefficients descriptors and time–related dedicated parameters [7].

The list of parameters with descriptions has been given in Table 1.

Table 1. Dataset details

no.	description
1	Temporal Centroid
2	Spectral Centroid average value
3	Spectral Centroid variance
4–37	Audio Spectrum Envelope (ASE) average values in 34 frequency bands
38	ASE average value (averaged for all frequency bands)
39–72	ASE variance values in 34 frequency bands
73	Averaged ASE variance parameters
74 – 75	Audio Spectrum Centroid – average and variance values
76 – 77	Audio Spectrum Spread – average and variance values
78–101	Spectral Flatness Measure (SFM) average values for 24 frequency bands
102	SFM average value (averaged for all frequency bands)
103–126	Spectral Flatness Measure (SFM) variance values for 24 frequency bands
127	Averaged SFM variance parameters
128–147	20 first mel cepstral coefficients average values
148–167	The same as 128–147
168–191	Dedicated parameters in time domain based of the analysis of the distribution of the envelope in relation to the rms value

3 Method Overview

The overall strategy was to address the music genre recognition problem using a mixed approach of a powerful classification model and multi–resolution clustering. The preliminary results, using a set of standard techniques like the Naive Bayes classifier, Multilayer Perceptron or cost–sensitive Random Forests, have never reached above 0.78 on the preliminary test set. Clearly some more sophisticated approach was required.

Examination of the training dataset using cluster analysis revealed that at an appropriate level of granularity, less than 5% of the clusters contained instances with mixed genres. This observation led to the assumption that each test set cluster would also predominantly represent a single genre.

The final approach thus involved multi–resolution clustering (Section 5) of the concatenated training and test sets, after labelling the latter with the

[3] Since parameters 128–147 and 148–167 are in fact the same, the effective size of the input space is 171.

classification model described in Section 4. A majority vote has then been taken for each cluster at the highest possible granularity level, which significantly improved the prediction quality. The new predictions were then fed back into the classification model and the whole process was repeated until convergence of the clusters.

For a diagram presenting an overview of our method please see Figure 1.

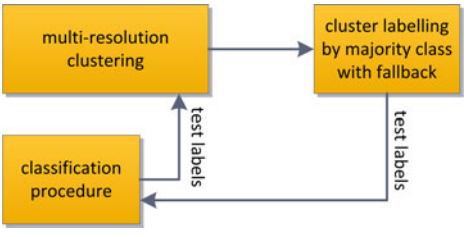


Fig. 1. Method overview

4 Error Correcting Output Coding (ECOC)

Error Correcting Output Coding (ECOC) is a technique for using binary classification algorithms to solve multi–class problems [3]. There are also other methods, which address the same issue, like the one–against–all strategy or decomposition of a c–class problem into $c(c-1)/2$ subproblems, one for each pair of classes [4]. Their primary focus however is to enable application of binary classifiers to multi–class problems, a very important function in the era of the omnipresent Support Vector Machines. The Error Correcting Output Coding has however some other interesting properties.

ECOC represents a distributed output code, in which each class is assigned a unique codeword – a binary string of a fixed length n. A set of n binary classifiers is then trained, one for each bit position in the codeword.

New instances are classified by evaluating all binary classifiers to generate a new n–bit string, which is then compared to all the codewords. The instance is assigned to the class whose codeword is closest to the generated string, where the similarity measure used is usually the Hamming distance [3], which for two binary strings s and t of length n is given by:

$$D_H(s,t) = \sum_{i=1}^{n} \left(1 - \delta_{s(i),t(i)}\right) \tag{1}$$

where $s(i)$ denotes the i^{th} bit of string s and δ_{ij} is the Kronecker delta function.

A distinguishing feature of ECOC is the way the codewords are generated. The most important property of ECOC codewords is that their pairwise Hamming distances are maximized. This results in a code which is able to correct as many individual classifier errors as possible. In general, a code with minimum pairwise

Hamming distance d is able to correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ individual bit (classifier) errors. Note, that the minimum Hamming distance in a code used by the one–against–all approach is 2, so no errors can be corrected in this case.

Properly generated ECOC codewords also have another useful property – they ensure that the individual classifiers are uncorrelated, that is each of them tends to make errors on different input instances. This results in an ensemble which is diverse, as all good ensembles should be [9].

4.1 Music Genre Classification System

For the six–class music genres dataset described in Section 2 an exhaustive ECOC has been used, leading to 31 two–class problems. The minimum pairwise Hamming distance between the codewords in this case is 16, which means that even if up to 7 individual classifiers make an incorrect prediction for a given input instance, the final prediction of the whole ensemble will still be correct.

The base classifier used was a LIBSVM [2] implementation of Support Vector Machine (SVM) [4] with linear kernels. The choice of this particular method has been dictated by preliminary experiments, which have revealed high resilience to over–fitting in the case of this particular dataset. The linear kernels have been used due to high dimensionality of the input space (so transforming it to an even higher dimensional space would be of negligible benefit), better scalability and reduced number of parameters that needed to be estimated, when compared to exponential or polynomial kernels. The parameters of each SVM have been estimated using a grid search approach within a 2–fold Density Preserving Sampling (DPS) scheme [1].

The training procedure has been depicted in Figure 2. Before training the data has been normalised to fit within the $(-1 \div 1)$ interval. The resultant predictive

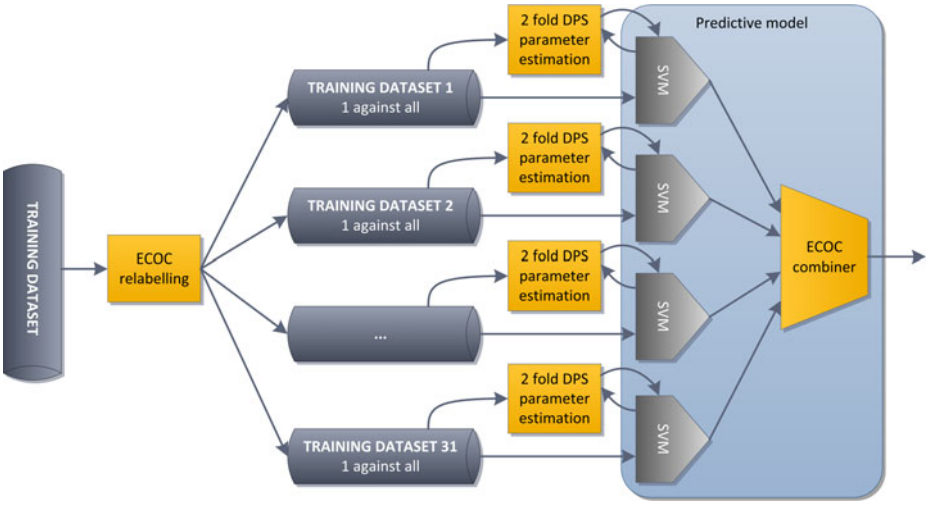


Fig. 2. Training of an Error Correcting Output Code classification system

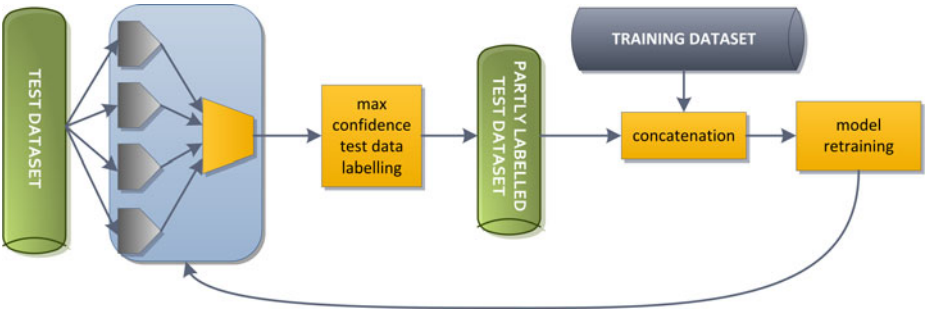


Fig. 3. Iterative semi–supervised learning

model consists of 31 SVMs and an ECOC combiner, responsible for matching the outputs of the individual classifiers to the closest codeword.

4.2 Iterative Semi–supervised Learning

An important feature of ensemble models is the possibility to assess the confidence of produced predictions by measuring the degree of disagreement between the ensemble members [10]. By taking advantage of this property it is possible to label the test dataset in an iterative manner, assigning the predicted labels only to the instances for which the ensemble confidence is above some threshold value. The part of the test dataset which has been labelled in this way is then appended to the training dataset, a new predictive model is built according to the procedure described in Section 4.1, and the whole process is repeated until convergence. This semi–supervised learning approach [5] has been depicted in Figure 3. If at some iteration no instances can be labelled with the required confidence, current predictions are taken as final and the process terminates.

5 Multi–resolution Clustering

The idea for the clustering came from experimenting with the data statistics, number of performers, number of genres and approximate number of songs. No crisp clusters were identified; however, we noticed that if a training instance was of a different genre from the rest of the cluster then it usually belonged to a different lower granularity cluster. The training and test data were first normalised and the R statistical software's [8] K–means clustering algorithm [4] implementation was used to assign the instances to clusters. It was then decided to "stretch out" the instances by introducing fine granularity clusters. The values of K we have used have been given in Table 2, alongside the rationale for choosing each particular value.

To validate the use of the multi–resolution clustering, the cluster assignation nominal labels were used to create a new dataset for the training and test data. WEKA [6] was used to run a Naive Bayes classifier on the training nominal

Table 2. Numbers of clusters in the K–means algorithm

K	rationale
6	Number of genres
15	Min number of music pieces per performer
20	Max number of music pieces per performer, number of segments per piece
60	Number of performers
300	Min number of music pieces per performer times number of segments
400	Max number of music pieces per performer times number of segments
600	As for 300, but with higher granularity
800	As for 400, but with higher granularity
900	Min number of music pieces in the dataset
1050	Mean number of music pieces in the dataset
1200	Max number of music pieces in the dataset
2000	As for 1000, but with higher granularity
3000	Higher granularity
3200	Higher granularity
5000	Higher granularity
7000	Higher granularity

cluster labels and the resulting model was applied to the test set. Submitting the results produced 0.7645 accuracy on the preliminary test set which we considered as validation of the clustering approach.

6 Iterative Relabelling Scheme

The relabelling scheme has been performed in two stages. The first stage was to attempt to relabel the whole cluster at the finest level of granularity. This may be achieved in two ways:

- If the cluster had a training instance member and 50% of the predictions had the same label as this training instance then the whole cluster was relabelled to match the training instance
- If the cluster had a 75% majority of prediction labels then the whole cluster was relabelled to the majority label.

This approach has been depicted in Figure 4.

If the cluster could not be relabelled at the highest, 7000 level by this approach then each instance in the cluster was checked in lower granularity clusters starting at the 5000 level. This time the instance is relabelled according to the diagram in Figure 5.

This process is analogous to the previous approach: if there is a training instance in the lower granularity cluster and 50% of the predictions in that cluster match the training instance then the test instance is relabelled to match the training instance. If there is no training instance in the lower granularity cluster and 75% of the predictions are one genre then the test instance is relabelled to

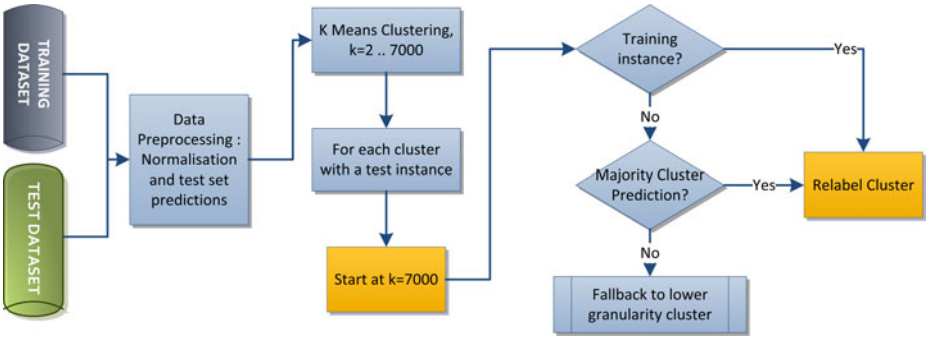


Fig. 4. Cluster relabelling

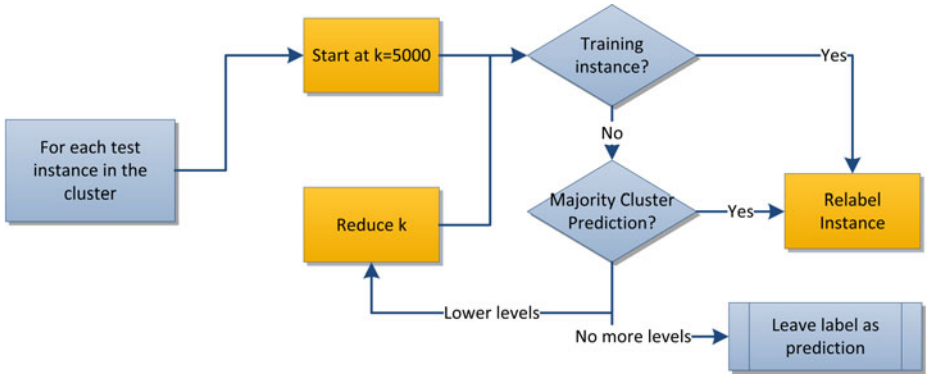


Fig. 5. Instance relabelling with fallback to lower granularity level

Table 3. Numbers of instances resolved by method and cluster

method/cluster	number of instances
Majority Prediction @ cluster 7000	8349
Majority Prediction @ cluster 5000	640
Majority Prediction @ cluster 3200	309
Majority Prediction @ cluster 3000	13
Training Instance @ cluster 7000	537
Training Instance @ cluster 5000	77
Training Instance @ cluster 3200	1
Left as predictions	343

this genre. If there is no resolution at the 5000 level then granularity is reduced again and the process is repeated. When all cluster levels have been checked using fallback and no consensus is reached then the test instance is left with its original predicted label.

The numbers of instances in the final submission resolved by the two strategies described above have been given in Table 3. As it can be seen, in practice the algorithm has never fallen back below the 3000 cluster level, so quite a fine level of granularity was needed for the method to provide reasonable performance.

7 Experimental Results

The final classification performance of the method proposed in this paper, as well as the performances achieved at particular steps of the development process, have been given in Table 4. Note, that the results for the final test set are given only, if a particular submission was final.

Table 4. Classification performance

Method/step	Preliminary performance	Final performance
0. Random Forest	0.7744	–
1. ECOC	0.8159	0.81521
2. Cluster relabelling	0.8558	–
3. ECOC (retrained)	0.8784	–
4. Final	0.8815	0.87507

As it can be seen, the first significant performance boost of around 3.5% on the preliminary test set came from replacing the Random Forest method with the semi–supervised ECOC ensemble. The next step, cluster relabelling with fallback to lower granularity level, has allowed to increase the classification accuracy by another 4%. The next major improvement of approximately 2.3% came from feeding the labels produced by cluster relabelling back to the ECOC ensemble. The final improvement was a result of another application of cluster relabelling. Since this improvement was rather modest, we have interpreted it as a sign of saturation of our method.

One of the strengths of the proposed method is that it reinforces correct predictions; however this may also be seen as one of the weaknesses as it reinforces the errors. From looking at the actual final test labels, we have incorrectly predicted whole clusters rather than individual instances. This has been particularly noticeable when discriminating between Rock and Metal, and Classical and Jazz.

8 Conclusions

The music genre classification method described in this paper has proven to be a well–performing and competitive solution. The method has been validated within the ISMIS 2011 Contest: Music Information Retrieval and has been ranked as a top solution, outperforming some 12 000 submission by almost 300 teams from all over the world, and achieving the classification rate of almost 88%.

This encouraging result inspired us to continue our research along this direction, using a diverse pool of benchmark and real datasets, which is currently an ongoing work.

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 251617.

References

1. Budka, M., Gabrys, B.: Correntropy–based density–preserving data sampling as an alternative to standard cross–validation. In: Proceedings of the IEEE World Congress on Computational Intelligence, pp. 1437–1444. IEEE, Los Alamitos (2010)
2. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
3. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Arxiv preprint cs/9501101 (1995)
4. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley & Sons, New York (2001)
5. Gabrys, B., Petrakieva, L.: Combining labelled and unlabelled data in the design of pattern classification systems. International Journal of Approximate Reasoning 35(3), 251–273 (2004)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
7. Kostek, B., Kupryjanow, A., Zwan, P., Jiang, W., Ras, Z., Wojnarski, M., Swietlicka, J.: Report of the ISMIS 2011 Contest: Music Information Retrieval. In: ISMIS 2011. Springer, Heidelberg (2011)
8. R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria (2006) ISBN 3-900051-07-0, http://www.R-project.org
9. Ruta, D.: Classifier diversity in combined pattern recognition systems. Ph.D. dissertation, University of Paisley (2003)
10. Schapire, R., Freund, Y., Bartlett, P., Lee, W.: Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics 26(5), 1651–1686 (1998)

Multi-label Learning Approaches for Music Instrument Recognition

Eleftherios Spyromitros Xioufis, Grigorios Tsoumakas, and Ioannis Vlahavas

Department of Informatics, Aristotle University of Thessaloniki, 54124 Greece
{espyromi,greg,vlahavas}@csd.auth.gr

Abstract. This paper presents the two winning approaches that we developed for the instrument recognition track of the ISMIS 2011 contest on Music Information. The solution that ranked first was based on the Binary Relevance approach and built a separate model for each instrument on a selected subset of the available training data. Moreover, a new ranking approach was utilized to produce an ordering of the instruments according to their degree of relevance to a given track. The solution that ranked second was based on the idea of constraining the number of pairs that were being predicted. It applied a transformation to the original dataset and utilized a variety of post-processing filters based on domain knowledge and exploratory analysis of the evaluation set. Both solutions were developed using the Mulan open-source software for multi-label learning.

1 Introduction

With the explosion of multimedia Internet services, such as YouTube and Last.fm, vast amounts of multimedia data are becoming available for exchange between Internet users. Efficiently browsing and searching in such enormous digital databases requires effective indexing structures. However, content-based indexing of multimedia objects such as music tracks is commonly based on manual annotations, since automatic understanding and categorization is still too difficult for computers. Here we focus on the challenging problem of recognizing pairs of instruments playing together in a music track.

While the automatic recognition of a single instrument is fairly easy, when more than one instruments play at the same time in a music track, the task becomes much more complex. The goal of the competition was to build a model based on training data concerning both single instruments and instrument mixtures in order to recognize pairs of instruments. The learning task can be viewed as a special case of multi-label classification [5], where the output of the classifier must be a set of exactly two labels.

Multi-label classification extends traditional single-label classification in domains with overlapping class labels (i.e. where instances can be associated with more than one labels simultaneously). More formally, in multi-label classification $\mathcal{X} = \mathbb{R}^M$ denotes the input attribute space. An *instance* $\boldsymbol{x} \in \mathcal{X}$ can

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 734–743, 2011.

be represented as an M-vector $\boldsymbol{x} = [x_1, \ldots, x_M]$. The set of L possible labels $Y = \{1, \ldots, L\}$ for a particular instance is represented by an L-vector $\mathbf{y} = [y_1, \ldots, y_L] = \{0, 1\}^L$ where $y_j = 1$ iff the jth label is relevant ($y_j = 0$ otherwise). A multi-label classification algorithm accepts as input a set of multi-label training examples $(\boldsymbol{x}, \mathbf{y})$ and induces a model that predicts a set of relevant labels for unknown test instances. In the domain of the contest, music instruments represent the overlapping class labels and each test instance is associated with exactly two of them.

Both solutions presented in this paper are based on the simple Binary Relevance (BR) approach for multi-label classification. BR learns L binary classifiers, one for each different label in Y. It transforms the original data set into L datasets that contain all examples of the original dataset, labeled positive if the original example was annotated with y_j and negative otherwise. For the classification of a new instance, BR outputs the union of the labels y_j that are positively predicted by the L classifiers.

The quality of predictions were evaluated by the contest organizers as follows:

- If no recognized instrument matched the actual ones, the score was 0
- If only one instrument was correctly recognized, the score was 0.5
- If both instruments matched the target ones, the score was 1.0

The final score of a solution was the average of its scores across all test instances.

The rest of the paper is organized as follows. Section 2 presents our exploratory analysis of the datasets. Sections 3 and 4 describe in detail the two solutions that we developed. Finally, Section 5 concludes this paper.

2 The Data

2.1 The Training Sets

The training set consisted of two datasets: one containing data of single instruments and one containing data of mixtures of instrument pairs. A first challenge of the contest, was that these two datasets were significantly heterogeneous.

The single instrument data comprised 114914 recordings of 19 different instruments. The instrument pairs data comprised just 5422 mixtures of 21 different instruments. In total there were 32 distinct instruments, just 8 of which appeared in both datasets. Table 1 presents the number and percentage of examples from each instrument in each of the two training datasets as well as in their union.

The mixtures dataset contained the following 12 different pairs of 21 instruments: (SopranoSaxophone, TenorTrombone), (AltoSaxophone, TenorTrombone), (TenorSaxophone, Tuba), (TenorSaxophone, B-FlatTrumpet), (BaritoneSaxophone, CTrumpet), (BassSaxophone, CTrumpet), (AcousticBass, Piano), (B-flatclarinet, Viola), (Cello, Oboe), (ElectricGuitar, Marimba), (Accordian, DoubleBass), (Vibraphone, Violin).

Table 1. Instrument distribution in the training sets and the test set

	Single		Pairs		Total		Validation		Test	
Instrument	Examples	%	Examples	%	Examples	%	Examples	%	Examples	%
SynthBass	918	0.7	0	0	918	1	595	4.0	635	4.3
EnglishHorn	1672	1.4	0	0	1672	1	0	0.0	0	0.0
Frenchhorn	2482	2.1	0	0	2482	2	364	2.4	425	2.9
Piccolo	2874	2.5	0	0	2874	2	0	0.0	0	0.0
Saxophone	4388	3.8	0	0	4388	3	0	0.0	0	0.0
Trombone	4503	3.9	0	0	4503	4	0	0.0	0	0.0
Bassoon	5763	5.0	0	0	5763	5	0	0.0	0	0.0
Flute	7408	6.4	0	0	7408	6	0	0.0	0	0.0
Clarinet	9492	8.2	0	0	9492	8	0	0.0	0	0.0
Trumpet	11152	9.7	0	0	11152	9	0	0.0	0	0.0
Guitar	34723	30.2	0	0	34723	28	0	0.0	0	0.0
Vibraphone	0	0	249	4.6	249	0	622	4.2	594	4.1
SopranoSaxophone	0	0	329	6.1	329	0	586	4.0	542	3.7
AltoSaxophone	0	0	337	6.2	337	0	405	2.7	388	2.6
B-FlatTrumpet	0	0	412	4	412	0	0	0.0	0	0.0
AcousticBass	0	0	417	4	417	0	0	0.0	0	0.0
BassSaxophone	0	0	503	9.2	503	0	340	2.4	335	2.4
BaritoneSaxophone	0	0	530	9.8	530	0	346	2.3	296	2.0
B-flatclarinet	0	0	567	10.5	567	0	3528	24.1	3661	25.0
ElectricGuitar	0	0	590	10.9	590	0	4149	28.3	4222	28.8
Marimba	0	0	590	10.9	590	0	455	3.1	377	2.6
TenorTrombone	0	0	666	12.2	666	1	1293	8.8	1248	8.5
TenorSaxophone	0	0	934	17.2	934	1	2137	14.6	2160	14.7
CTrumpet	0	0	1033	19.0	1033	1	2155	14.7	2048	14.0
Oboe	1643	1.4	332	6.1	1975	2	3184	21.7	3247	22.1
Accordian	1460	1.2	634	11.7	2094	2	2466	16.8	2427	16.6
Viola	3006	2.6	567	10.4	3573	3	762	5.2	815	5.6
Tuba	3463	3.0	522	9.6	3985	3	0	0	0	0.0
DoubleBass	3849	3.3	634	11.7	4483	4	1384	9.4	1338	9.1
Violin	5010	4.4	249	4.6	5259	4	3528	22.2	3237	22.1
Cello	4964	4.3	332	6.1	5296	4	698	4.7	694	4.7
Piano	6144	5.3	417	7.7	6561	5	595	4.0	635	4.3

It is interesting to notice that the pairs dataset contained instruments that can be considered as *kinds* of instruments in the single instruments dataset. SopranoSaxophone, AltoSaxophone, TenorSaxophone, BaritoneSaxophone and BassSaxophone are kinds of Saxophone, CTrumpet and B-FlatTrumpet are kinds of Trumpet, TenorTrombone is a kind of Trombone, B-FlatClarinet is a kind of Clarinet and ElectricGuitar is a kind of Guitar. These relations complicate the learning problem in some ways. Firstly, examples of the specialized class (e.g. TenorTrombone) could be semantically considered as examples of the general class (e.g. Trombone). It may be difficult to distinguish between such parent-child pairs of classes. Secondly, different kinds of the same instrument could be difficult to distinguish (e.g. is one of the instruments a soprano or an alto saxophone?).

It is also interesting to notice a special property of the *ElectricGuitar* and *Marimba* instruments. It is the only pair of instruments that satisfies both of the following conditions: a) none of the instruments of the pair appears in the single instruments dataset, b) none of the instruments of the pair appears together with another instrument in the mixtures dataset. This means that out of the 32 instruments, these particular two instruments are the only ones to have exactly the same positive and negative training examples. It would therefore be impossible for any classifier to distinguish them.

2.2 The Test Set

Besides the heterogeneity of the training sets, the following statements about the synthesis of the test set brought additional complexity to the learning task:

- Test and training sets contain different pairs of instruments (i.e. the pairs from the training set do not occur in the test set).
- Not all instruments from the training data must also occur in the test part.
- There may be some instruments from the test set that only appear in the single instruments part of the training set.

In order to have a clearer idea about the synthesis of the test set and for other reasons which will be explained in the analysis of the respective solutions, we queried the evaluation system for the frequency of each instrument in the test set by submitting a prediction containing the same instrument for all test instances. Table 1 (Column 4) contains the percentage of each label measured in the validation set (35% of the test data) along with a projection of the expected number of examples in the full test set. Column 5 of Table 1 contains the actual percentage and number of examples of each label in the full test set.

By examining Table 1, we reach to the following conclusions:

- Only 20 out of the 32 instruments appear in the test set.
- The mixtures training set contained 18 of the 20 instruments of the test set plus 3 additional instruments.
- The single instruments training set contained 9 of the 20 instruments of the test set plus 10 additional instruments.
- There is a great discrepancy between the distribution of the labels in the training and the test data.

2.3 Features

Each track in both the training and the test data was described by 120 precomputed attributes capturing various sound properties:

- Flatness coefficients: BandsCoef1-33, bandsCoefSum
- MFCC coefficients: MFCC1-13
- Harmonic peaks: HamoPk1-28
- Spectrum projection coefficients: Prj1-33, prjmin, prjmax, prjsum, prjdis, prjstd
- Other acoustic spectral features: SpecCentroid, specSpread, energy, log spectral centroid, log spectral spread, flux, rolloff, zerocrossing
- Temporal features: LogAttackTime, temporalCentroid

The single instruments set was described by the following additional five attributes:

- Frameid - Each frame is 40ms long signal
- Note - Pitch information
- Playmethod - One schema of musical instrument classification according to the way they are played
- Class1,class2 - Another schema of musical instrument classification according to Hornbostel-saches

3 Investigation of Multi-label Learning Methods

A first important issue was to determine which multi-label learning method was the most appropriate one for our particular problem. We compared the performance of various state-of-the-art multi-label methods that were available in our Mulan open-source software for multi-label learning[1], such as ECC [3], CLR [2] and RAKEL [6] along with baseline methods such as the Binary Relevance (BR) approach. In this first set of experiments the union of the two datasets (single and mixtures) was used as the training set and the performance of the methods was evaluated directly on the test set. The reason was that the training data was substantially different from the test data (see Section 2) and the results of a comparison on the training data could be misleading.

The results of a comparison using various binary base classifiers revealed that state-of-the-art multi-label methods had little or no benefit in comparison with the simple BR approach, especially when BR was coupled with ensemble-based binary classifiers such as Random Forest [1]. The results were not surprising since the main advantage of advanced multi-label learning methods over the BR approach is their ability to capture and exploit correlations between labels. In our case, learning the correlations which appear in the training set was not expected to be useful since these correlations are not repeated in the test set.

4 The Solution that Ranked First

4.1 Engineering the Input

While in our initial set of experiments we used the union of the given training sets, we were also interested in measuring the performance of the methods given either only the mixture or only the single-instrument examples as training data. The results showed that using only the mixture examples for training was far better than using only the single-instrument examples, and was even better than using all the available training examples. We gave two possible explanations for this outcome:

- Learning from pairs of instruments is better when the task is to predict pairs of instruments (even though the pairs appearing in the test set are different).
- The distribution of the labels in the mixtures dataset matches better to that of the test set.

The findings regarding the nature of the test set, presented in Subsection 2.2, were quite revealing. By using only the single-instruments set for training, we could predict only 9 of the 20 instruments which appear in the test set, compared to 18 when using the mixtures set. However, it was still difficult to determine why using the mixtures set alone was better than combining all the data since, in the latter case, all the relevant instruments were present in the training set. To make things more clear we performed a new set of experiments.

[1] mulan.sourceforge.net

We first removed the training data corresponding to the 12 instruments which were not present in the test set and then created the following training sets: a) One that contained both mixture and single-instrument examples for the instruments appearing in the test set. b) One that contained only mixture examples for the 18 out of 20 instruments and single-instrument examples for the 2 remaining instruments of the test set. c) One that contained only single-instrument examples for the 9 out of 20 instruments and mixture examples for the rest 11 instruments of the test set. The best results were obtained using the second training set, and verified that learning from mixtures is better when one wants to recognize mixtures of instruments. Note that adding single-instrument examples for the 2 instruments which had no examples in the mixtures set, slightly improved the performance of using only examples of mixtures. This revealed that using single-instrument data can be beneficial in the case that no mixture data is available. The set used to train the winning method comprised of the union of the 5422 mixture examples and the 340 single-instrument examples of SynthBass and Frenchhorn. All the given feature attributes describing the mixture examples were used, while we ignored the 5 additional attributes of the single-instruments set since they were not present in the test set.

4.2 Base Classifier

A problem arising from the use of the one-versus-rest or BR approach for multi-label classification is that most of the labels have much more negative than positive examples. Class imbalance is known to negatively affect the performance of classifiers by biasing their focus towards the accurate prediction of the majority class. This often results in poor accuracy for the minority class, which is the class of interest in our case. For this reason, special attention was paid on selecting a classification scheme that is able to tackle this problem.

To deal with class imbalance we extended the original Random Forest (RF) [1] algorithm. RF creates an ensemble of unpruned decision trees where each tree is built on a bootstrap sample of the training set. Random feature selection is used in the tree induction process. To predict the class of an unknown object the predictions of the individual trees are aggregated. RF has been proven to have superior accuracy among current classification algorithms, however, it is susceptible on imbalanced learning situations. Our idea is based on combining RF with Asymmetric Bagging [4]. Instead of taking a bootstrap sample from the whole training set, bootstrapping is executed only on the examples of the majority (negative) class. The Asymmetric Bagging Random Forest (ABRF) algorithm is given below:

1. Take a sample with replacement from the negative examples with size equal to the number of positive examples. Use all the positive examples and the negative bootstrap sample to form the new training set.
2. Train the original RF algorithm with the desired number of trees on the new training set.
3. Repeat the two steps above for the desired number of times. Aggregate the predictions of all the individual *random trees* and make the final prediction.

Building a forest of 10 random trees on each one of 10 balanced training sets yielded the best evaluation results.

4.3 Informed Ranking

The output produced for each label by an ABRF classifier can be used either as a hard classification (the decision of the majority) or transformed into a confidence score of the label being true by dividing the number of random trees that voted for the label with the total number of random trees. In a typical multi-label classification problem (where the number of relevant labels for each test instance is unknown) we would either use the first approach to select the relevant labels for each test instance, or apply a decision threshold to the confidence scores in order to transform them into hard classifications. In the domain of the contest though, we a priori knew that exactly two instruments are playing on each track, thus we followed a different approach. We focused on producing an accurate ranking of the labels according to their relevance to each test instance and selected the two top-ranked labels. Instead of directly using the confidence scores to produce a ranking of the labels, we developed a novel ranking approach which takes into account the prior probability distribution of the labels. Our approach is as follows:

1. Use the trained classifiers to generate confidence scores for all test instances.
2. Sort the list of confidence scores given for each label.
3. Given a test instance, find its rank in the sorted list of confidences for each label. These ranks are indicative of the relevance of the instance to each label.
4. Normalize the ranks produced from step 3 by dividing them with the estimated (based on their prior probabilities) number of relevant instances for each label in the test set and select the n labels with the lowest normalized rank.

We explain the effect of normalization with an example: Assume that we have 100 test instances and an instance x_i is ranked 30th for label1 and label2 and 40th for label3. We further know that only one label is relevant for x_i and that the prior probabilities of the labels are $P(label1) = P(label2) = 0.25$ and $P(label3) = 0.5$. By normalizing the ranks we get 30/25 for label1 and label2 and 40/50 for label3. Thus, we would select label3 for x_i although label1 and label2 have a lower absolute rank. This is rational since based on the priors we expect that label1 and label2 will have only 25 relevant instances and x_i's rank for these labels was 30. In the context of the contest, we had the chance to use the frequencies of the labels in the validation set to estimate the number of relevant instances in the full test set. In a real-world situation, the prior probabilities of the labels in the training set could be used for this purpose.

4.4 Engineering the Output

As a final step, a post-processing filter was applied which disallowed instrument pairs that were present in the training set. In such cases, the second-ranked label

was substituted by the next label which would not produce a label pair of the training set when combined with the first-ranked label. This substitution was based on the assumption that the classifier is more confident for the first-ranked label. The information for this filter was given in the description of the task by the contest organizers (see Section 2).

5 The Solution that Ranked Second

The mixtures dataset consists of 5422 examples, yet the number of distinct instrument pairs it contains is just 12. This observation, led us to the hypothesis that the test set, which consists of 14663 instances, might also contain a small number of instrument pairs. However, the number of distinct instrument pairs predicted by our early attempts on the problem was quite large. This led to the core idea of this solution: constraining the number of pairs that were being predicted.

5.1 Engineering the Input

A first step was to join the two training datasets into a single one. The extra features of the single-instruments dataset were deleted in this process, while the label space of the datasets was expanded to cover the union of all labels. The union of the examples of the two datasets was then considered.

We then adopted the following transformation of this dataset. We considered a new label space consisting of all pairs of instruments. The labels of this new label space had a positive value, whenever one of the labels in the original space, i.e. one of the instruments, had a positive value. In other words, the new label space applied an OR operator on all pairs of the original labels space. Figure 1 exemplifies this process with just three instruments, respecting the pairs that appear in the training set.

Cello	Oboe	Piano		Cello OR Oboe	Cello OR Piano	Oboe OR Piano
true	true	false		true	true	true
true	false	false	⇒	true	true	false
false	true	false		true	false	true
false	false	true		false	true	true

Fig. 1. Transformation of the data to a new label space

This quite strange transformation was motivated from the fact that the task required us to predict pairs of instruments, but didn't provide us with examples of mixtures of these pairs. The transformation allowed the direct modeling of all pairs, using as examples either available mixtures, or available examples of one of the two instruments.

5.2 Learning

We applied the binary relevance approach on the transformed dataset. Each of the binary models was trained using the random forest algorithm [1] with 200 trees, after random sub-sampling so as to have at most a 10:1 ratio between the negative and positive class. Given a test instance, the output of this approach was a ranking of the labels (pairs of instruments) according to relevance to each of the test instances, based on the probability estimates of the random forest algorithm.

5.3 Engineering the Output

As already mentioned in the beginning of this section, the key point of this solution was constraining the instrument pairs given in the output. This was achieved via a variety of filters operating at a post-processing step after the learning step has ranked all possible pairs of instruments.

A first simple post-processing filter disallowed instrument pairs that were present in the training set. The information for this filter was given in the description of the task by the contest organizers (see Section 2). A second filter disallowed instrument pairs, where at least one of the instruments was absent from the evaluation set, as discovered from our exploratory analysis of the evaluation set (see Section 2).

Then a number of filters were applied, one for each instrument that was present in the evaluation set, which disallowed pairs of this instrument with other instruments based on two main information sources:

- Domain knowledge, which was sought in the Internet, as our musical literacy was rather limited for this task. The Vienna Symphonic Library[2] was a good source of knowledge for combinations of instruments that make sense. We also issued Google queries for pairs of instruments and considered the number of returned documents as evidence supporting the common appearance of these instruments in a music track. Sites with free music pieces for instruments were also consulted.
- The projected instrument distribution in the test set based on the evaluation set (see Section 2). Instruments with predicted distribution much higher than the projected one, hinted us that pairs containing them should be candidates for removal from the allowed set of instrument pairs. On the other hand, instruments with predicted distribution much lower than the projected one, hinted us that perhaps we have wrongly disallowed pairs containing them.

Constructing this last set of filters was a time-consuming iterative process, involving several submissions of results for evaluation feedback, that in the end led to allowing just 20 instrument pairs. After the test set was released, we found out that it actually contained 24 instrument pairs, 13 of which were within the allowed 20 by our approach. The remaining 11 were disallowed by our approach, which further allowed 7 pairs that were not present in the test set.

[2] http://www.vsl.co.at/

Instrument pairs were examined in the order of relevance to a test instance as output by the learning algorithm, until a pair that was not disallowed by the filters was reached. This was the final output of the post-processing algorithm for that test instance.

We also included another post-processing step that led to slight improvements. This step took into account the parent-child relationships of instruments that were discussed in Section 2 and performed the following replacements of instruments in a predicted pair, prior to passing this pair from the filters: Clarinet was replaced by B-FlatClarinet, Trumpet by CTrumpet, Guitar by ElectricGuitar and Trombone by TenorTrombone.

6 Conclusions

Our motivation for participating in the instrument recognition track of the ISMIS 2011 Contest on Music Information Retrieval was to explore the potential of multi-label learning methods [5].

One interesting conclusion was that in multi-label learning problems, like the one of this contest, where modeling label correlations is not useful, combining simple multi-label learning techniques, such as Binary Relevance with strong single-label learning techniques, such as Random Forest, can lead to better performance compared to state-of-the-art multi-label learning techniques. Another interesting conclusion derived from the solution that ranked first was that it is better to use only mixture examples when pairs of instruments need to be recognized.

An interesting direction for the next year's contest would be the generalization of the task to the recognition of an arbitrary number of instruments playing together.

References

1. Breiman, L.: Random forests. Mach. Learn. 45, 5–32 (2001)
2. Fürnkranz, J., Hüllermeier, E., Mencia, E.L., Brinker, K.: Multilabel classification via calibrated label ranking. Machine Learning (2008)
3. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: Proceedings of ECML PKDD 2009, Bled, Slovenia, pp. 254–269 (2009)
4. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1088–1099 (2006)
5. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, 2nd edn., ch. 34, pp. 667–685. Springer, Heidelberg (2010)
6. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multi-label classification. IEEE Transactions on Knowledge and Data Engineering (2011)

Author Index