

Mykola Pechenizkiy
Marek Wojciechowski (Eds.)

New Trends in Databases and Information Systems

 Springer

Editor-in-Chief

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

Mykola Pechenizkiy and Marek Wojciechowski (Eds.)

New Trends in Databases and Information Systems

 Springer

Editors

Mykola Pechenizkiy
Department of Computer Science
Eindhoven University of Technology
Eindhoven
The Netherlands

Marek Wojciechowski
Institute of Computing Science
Poznan University of Technology
Poznan
Poland

ISSN 2194-5357

ISBN 978-3-642-32517-5

DOI 10.1007/978-3-642-32518-2

Springer Heidelberg New York Dordrecht London

e-ISSN 2194-5365

e-ISBN 978-3-642-32518-2

Library of Congress Control Number: 2012943846

© Springer-Verlag Berlin Heidelberg 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

Preface

Database and information systems technologies have been rapidly evolving in several directions over the past years. New types and kinds of data, new types of applications and information systems to support them raise diverse challenges to be addressed. The so-called big data challenge, streaming data management and processing, social networks and other complex data analysis, including semantic reasoning into information systems supporting for instance trading, negotiations, and bidding mechanisms are just some of the emerging research topics. This volume contains papers contributed by six workshops and the Ph.D. Consortium associated with the ADBIS 2012 conference that report on the recent developments and an ongoing research in the aforementioned areas.

The 16th East-European Conference on Advances in Databases and Information Systems (ADBIS 2012) was held on September 17–21, 2012 in Poznan, Poland. The main objective of the ADBIS series of conferences is to provide a forum for the dissemination of research accomplishments and to promote interaction and collaboration between the database and information systems research communities from Central and East European countries and the rest of the world. The general idea behind the workshops and the Ph.D. Consortium was to collect contributions from various subdomains of the broad research areas of databases and information systems, representing new trends in these two important areas. Each of the events complementing the main ADBIS conference had its own international program committee, whose members served as the reviewers of papers contributed to the corresponding part of this volume.

ADBIS Workshop on GPUs in Databases (GID 2012) was chaired by Witold Andrzejewski (Poznan University of Technology, Poland), Krzysztof Kaczmarski (Warsaw University of Technology, Poland), and Tobias Lauer (Jedox AG, Germany). The motivation behind the workshop was an observation that while other application domains strongly benefit from utilizing the Graphics Processing Units (GPUs) to increase the performance of processing, database-related applications and algorithms do not get enough attention. The main goal of GID 2012 was to fill this gap. Members of the workshop's Program Committee were: Amitava Datta (University of Western Australia, Commonwealth of Australia), Jarosław Gramacki

(University of Zielona Góra, Poland), Bingsheng He (Nanyang Technological University, Singapore), Ming Ouyang (University of Louisville, USA), John D. Owens (University of California, Davis, USA), Krzysztof Stencel (Warsaw University of Technology, Poland), and Paweł Wojciechowski (Poznan University of Technology, Poland).

Mining Complex and Stream Data (MCSD'12) was chaired by Jerzy Stefanowski (Poznan University of Technology, Poland) and Dominik Ślęzak (University of Warsaw & Infobright Inc., Poland). The aim of this workshop was introducing new algorithmic foundations and application aspects of mining real world difficult data with a particular focus on the emerging domain of stream data mining and mining large volumes of data having a complex structure. The Program Committee of MCSD'12 consisted of the following members: Jan Bazan (University of Rzeszów, Poland), Petr Berka (University of Economics, Prague, Czech Republic), Albert Bifet (University of Waikato, New Zealand), Michelangelo Ceci (University of Bari, Italy), Alfredo Cuzzocrea (ICAR-CNR & University of Calabria, Italy), Simon Fischer (Rapid-I GmbH, Dortmund, Germany), Mohamed Gaber (University of Portsmouth, UK), Jerzy Grzymała-Busse (University of Kansas, USA), Rudolf Kruse (Otto-von-Guericke University, Magdeburg, Germany), Marzena Kryszkiewicz (Warsaw University of Technology, Poland), Stan Matwin (University of Ottawa, Canada), Ernestina Menasalvas (Technical University of Madrid, Spain), Mikołaj Morzy (Poznan University of Technology, Poland), Hung Son Nguyen (The University of Warsaw, Poland), Zbigniew Raś (University of North Carolina, Charlotte, USA), Alexey Tsymbal (Siemens AG, Erlangen, Germany), Michał Woźniak (Wrocław University of Technology, Poland), and Indre Zliobaite (Bournemouth University, UK). The reviewing process was also supported by Dariusz Czerski (IPI PAN at Warsaw, Poland), Piotr Sobolewski (Wrocław University of Technology, Poland), and Gianvito Pio (University of Bari, Italy).

International Workshop on Ontologies meet Advanced Information Systems (OAIS'2012) was chaired by Ladjel Bellatreche (LIAS/ISAE-ENSMA, France) and Yamine Ait Ameer (IRIT-ENSEEIH, France). The workshop had two main objectives. The first one was to present new and challenging issues in the contribution of ontologies for designing high quality information systems. The second one was to present new research and technological developments that use ontologies all over the life cycle of information systems. The Program Committee members of OAIS'2012 were: Brahim Medjahed (Michigan University, USA), David Taniar (Monash University, Australia), Oscar Romero Moral (Universitat Politècnica de Catalunya), Pascal Hitzler (Wright State University, USA), Carlos Ordóñez (Houston University USA), Dickson K.W. Chiu (University of Hong Kong, China), Francesco Guerra (Università di Modena e Reggio Emilia, Italy), Fernando Silva Parreiras (FUMEC University, Brazil), Dimitris Plexousakis (Crete University, Greece), Leandro Krug Wives (Federal University of Rio Grande do Sul, Brazil), Haridimos Kondylakis (FORTH-ICS and University of Crete, Greece), Reza Akbarinia (INRIA, Montpellier, France), Manolis Koubarakis (National and Kapodistrian University of Athens, Greece), Filipe Mota Pinto (Polytechnic Institute of Leiria, Portugal), Stéphane Jean (LIAS/ISAE-ENSMA, France), Mimoun Malki (Sidi Bel Abbès University,

Algeria), Boufaïda Zizette (Constantine University, Algeria), Abdelkamel Tari (Béjaïa University, Algeria), Farouk Toumani (Clermont Ferrand University, France), Chantal Reynaud (LRI, Paris, France), Alfredo Cuzzocrea (ICAR-CNR and University of Calabria, Italy), Daniela Grigori (Prism, Versailles University, France), Juan C. Trujillo (University of Alicante, Spain), Zohra Bellahsene (LIRMM, Montpellier, France), Shonali Krishnaswamy (Monash University, Australia), Selma Khouri (LIAS/ISAE-ENSMA, France), Idir Ait Sadoune (Supelec, France), and Simitsis Alkis (HP, USA).

Second Workshop on Modeling Multi-commodity Trade: Data models and processing (MMT'12) was chaired by Eugeniusz Toczyłowski (Warsaw University of Technology, Poland) and Mariusz Kaleta (Warsaw University of Technology, Poland). Its goal was to address the current scientific and technological challenges in information systems supporting the market mechanism, including: semantics issues in trading, architectures of information systems for market mechanisms, incorporating social networks into trade processes, data modeling for negotiations and bidding, as well as market processes modeling and management. The Program Committee of MMT'12 consisted of the following members: Stanisław Ambroszkiewicz (Polish Academy of Sciences, Poland), Costin Badica (University of Craiova, Romania), Janusz Granat (National Institute of Telecommunications, Poland), Przemysław Kazienko (Wrocław University of Technology, Poland), Zbigniew Nahorski (Polish Academy of Science, Poland), Marcin Paprzycki (Polish Academy of Science, Poland), and Adam Wierzbicki (Polish-Japanese Institute of Information Technology, Poland).

1st ADBIS Workshop on Social Data Processing (SDP'12) was chaired by Jaroslav Pokorný (Charles University in Prague, Czech Republic), Athena Vakali (Aristotle University of Thessaloniki, Greece), and Václav Snášel (VSB - Technical University of Ostrava, Czech Republic). The workshop aimed at addressing the research issues associated with online social networks, including the topics of distributed computing, databases, and storage systems as well as modeling professional profiles of objects like workers, specialists, projects, supervisors, etc. The members of the Program Committee of SDP'12 were: Peter Vojtáš (Charles University in Prague, Czech Republic), Kamil Matoušek (Czech Technical University, Czech Republic), Jiří Kubalík (Czech Technical University, Czech Republic), Petr Křemen (Czech Technical University, Czech Republic), Hakim Hacid (Bell Labs, France), Nick Papanikolaou (HP Labs, UK), Myra Spiliopoulou (University of Magdeburg, Germany), Ernestina Menasalvas (Technical University of Madrid, Spain), Maria Augusta Nunes (Federal University of Sergipe, Brazil), Miloš Kudělka (VSB - Technical University of Ostrava, Czech Republic), and Jan Martinovič (VSB - Technical University of Ostrava, Czech Republic).

1st ADBIS Workshop on Social and Algorithmic Issues in Business Support (SAIBS) was chaired by Adam Wojciechowski (Poznan University of Technology, Poland) and Alok Mishra (Atılım University, Turkey). The focus of the workshop was on computational and optimization issues that can be supported by crowd input or social intelligence. Specific goals included addressing the following problems: how far and on which fields business may benefit from utilizing social contribution,

and how computer systems may understand social behavior and support humans in making decisions. The Program Committee of SAIBS consisted of the following members: Ricardo Colomo Palacios (Carlos III University of Madrid, Spain), Arianna D'Ulizia (IRPPS, National Research Council, Rome, Italy), Fernando Ferri (IRPPS, National Research Council, Rome, Italy), Patrizia Grifoni (IRPPS, National Research Council, Rome, Italy), Kyoung Jun Lee (Kyung Hee University, Korea), Mirosław Ochodek (Poznan University of Technology, Poland), Rory O'Connor (Dublin City University, Ireland), Robert Susmaga (Poznan University of Technology, Poland), and Agnieszka Węgrzyn (University of Zielona Góra, Poland).

ADBIS Ph.D. Consortium was established as a forum for Ph.D. students to present their research ideas, confront them with the scientific community, and receive feedback from senior mentors. Ph.D. students at an advanced stage of research were given an opportunity to prepare a paper devoted to their research area with a possibility of sharing their achieved preliminary results. The chairs of the Ph.D. Consortium, responsible for selecting the papers from this category, were Mikołaj Morzy (Poznan University of Technology, Poland) and Alexandros Nanopoulos (Catholic University of Eichstätt-Ingolstadt, Germany).

We would like to thank the authors, the reviewers, and the chairs of the ADBIS 2012 workshops for their work and effort without which assembling this volume would not be possible.

September 2012

Mykola Pechenizkiy
Marek Wojciechowski

allegro group

POZnan*
*Miasto know-how

IBM

Microsoft®

Roche

T A R G I T  **®**
business intelligence

SAMSUNG

EDGE  **SOLUTIONS**

itelligence

Contents

Part I: GPUs in Databases

Applying CUDA Technology in DCT-Based Method of Query Selectivity Estimation	3
<i>Dariusz Rafal Augustyn, Sebastian Zederowski</i>	
Processing of Range Query Using SIMD and GPU	13
<i>Pavel Bednář, Petr Gajdoš, Michal Krátký, Peter Chovanec</i>	
Towards Optimization of Hybrid CPU/GPU Query Plans in Database Systems	27
<i>Sebastian Breß, Eike Schallehn, Ingolf Geist</i>	
Thrust and CUDA in Data Intensive Algorithms	37
<i>Krzysztof Kaczmarski, Paweł Rzążewski</i>	

Part II: Mining Complex and Stream Data

A Detection of the Most Influential Documents	49
<i>Dariusz Ceglarek, Konstanty Haniewicz</i>	
Approximation Algorithms for Massive High-Rate Data Streams	59
<i>Alfredo Cuzzocrea</i>	
Comparing Block Ensembles for Data Streams with Concept Drift	69
<i>Magdalena Deckert, Jerzy Stefanowski</i>	
Adapting Travel Time Estimates to Current Traffic Conditions	79
<i>Przemysław Gawęł, Krzysztof Dembczyński, Robert Susmaga, Przemysław Wesolek, Piotr Zielniewicz, Andrzej Jaszkievicz</i>	

SONCA: Scalable Semantic Processing of Rapidly Growing Document Stores	89
<i>Marek Grzegorowski, Przemysław Wiktor Pardel, Sebastian Stawicki, Krzysztof Stencel</i>	
Collective Classification Techniques: An Experimental Study	99
<i>Tomasz Kajdanowicz, Przemysław Kazienko, Marcin Janczak</i>	
Granular Knowledge Discovery Framework: A Case Study of Incident Data Reporting System	109
<i>Adam Krasuski, Dominik Ślęzak, Karol Kreński, Stanisław Łazowy</i>	
Diversity in Ensembles for One-Class Classification	119
<i>Bartosz Krawczyk</i>	
Evaluation of Stream Data by Formal Concept Analysis	131
<i>Martin Radvanský, Vladimír Sklenář, Václav Snášel</i>	
Soft Competitive Learning for Large Data Sets	141
<i>Frank-Michael Schleich, Xibin Zhu, Barbara Hammer</i>	
Enhancing Concept Drift Detection with Simulated Recurrence	153
<i>Piotr Sobolewski, Michał Woźniak</i>	
DeltaDens – Incremental Algorithm for On-Line Density-Based Clustering	163
<i>Radostaw Z. Ziemiński</i>	
Part III: Ontologies Meet Advanced Information Systems	
Introducing Artificial Neural Network in Ontologies Alignment Process	175
<i>Warith Eddine Djeddi, Mohamed Tarek Khadir</i>	
Time Integration in Semantic Trajectories Using an Ontological Modelling Approach: A Case Study with Experiments, Optimization and Evaluation of an Integration Approach	187
<i>Rouaa Wannous, Jamal Malki, Alain Bouju, Cécile Vincent</i>	
WebOMSIE: An Ontology-Based Multi Source Web Information Extraction	199
<i>Zineb Younsi, Mohamed Quafafou, Redouane Ouzegane, Abdelkamel Tari</i>	
Part IV: Modeling Multi-commodity Trade: Data Models and Processing	
Bidding Languages for Continuous Auctions	211
<i>Mariusz Kaleta</i>	

Auction of Time as a Tool for Solving Multiagent Scheduling Problems	221
<i>Piotr Modliński</i>	
Application of an Auction Algorithm in an Agent-Based Power Balancing System	231
<i>Piotr Pałka, Weronika Radziszewska, Zbigniew Nahorski</i>	
Multi-commodity Trade Application to the Routing Algorithm for the Delay and Disruptive Tolerant Networks	241
<i>Piotr Pałka, Radosław Schoeneich</i>	
Offers Discovery and Identifying User Requirements for Multi-commodity Trade in Open Markets	251
<i>Dominik Ryżko, Anna Wróblewska</i>	
Fair Resource Allocation in Multi-commodity Networks	261
<i>Tomasz Śliwiński</i>	
Part V: Social Data Processing	
Heuristic Approach to Automatic Wrapper Generation for Social Media Websites	273
<i>Bartosz Baziński, Michał Brzezicki</i>	
Spectral Clustering: Left-Right-Oscillate Algorithm for Detecting Communities	285
<i>Pavla Dráždilová, Jan Martinovič, Kateřina Slaninová</i>	
Exploiting Potential of the Professional Social Network Portal “SitIT”	295
<i>Kamil Matoušek, Jiří Kubalík, Martin Nečaský, Peter Vojtáš</i>	
Modeling and Storing Complex Network with <i>Graph-Tree</i>	305
<i>Adan Lucio Pereira, Ana Paula Appel</i>	
Evolution of Author’s Profiles Based on Analysis of DBLP Data	317
<i>Martin Radvanský, Zdeněk Horák, Miloš Kudělka, Václav Snášel</i>	
Towards Effective Social Network System Implementation	327
<i>Jaroslav Škrabálek, Petr Kunc, Filip Nguyen, Tomáš Pitner</i>	
Part VI: Social and Algorithmic Issues in Business Support	
Community Traffic: A Technology for the Next Generation Car Navigation	339
<i>Przemysław Gawel, Krzysztof Dembczyński, Wojciech Kotłowski, Marek Kubiak, Robert Susmaga, Przemysław Wesolek, Piotr Zielniewicz, Andrzej Jaszkievicz</i>	

Situational Requirement Method System: Knowledge Management in Business Support	349
<i>Deepti Mishra, Secil Aydin, Alok Mishra</i>	
Effectiveness Analysis of Promotional Features Used in Internet Auctions: Empirical Study	361
<i>Adam Wojciechowski, Paweł Warczynski</i>	
Part VII: Ph.D. Consortium	
Data Mining Approach to Digital Image Processing in Old Painting Restoration	373
<i>Joanna Gancarczyk</i>	
Determining Document's Semantic Orientation Using kNN Algorithm	383
<i>Krzysztof Jędrzejewski, Maurycy Zamorski</i>	
Designing a Software Transactional Memory for Peer-to-Peer Systems	395
<i>Aurel Paulovič, Pavol Návrat</i>	
Traceability in Software Architecture Decisions Based on Notes about Documents	403
<i>Gilberto Pedraza-Garcia, Dario Correal</i>	
OLAP Models for Sequential Data – Current State of Research and Open Problems	415
<i>Łukasz Nienartowicz</i>	
Data Management for Fingerprint Recognition Algorithm Based on Characteristic Points' Groups	425
<i>Michał Szczepanik, Ireneusz Józwiak</i>	
Data Prefetching Based on Long-Term Periodic Access Patterns	433
<i>Dmitri Vasilik</i>	
E-ETL: Framework for Managing Evolving ETL Processes	441
<i>Artur Wojciechowski</i>	
Author Index	451

Part I
GPUs in Databases

Applying CUDA Technology in DCT-Based Method of Query Selectivity Estimation

Dariusz Rafal Augustyn and Sebastian Zederowski

Abstract. The problem of efficient calculation of query selectivity estimation is considered in this paper. The selectivity parameter allows database query optimizer to estimate the size of the data satisfying given condition, which is needed to choose the best query execution plan. Obtaining query selectivity in case of a multi-attribute selection condition requires a representation of multidimensional attributes values distribution. This paper describes in details solution of this problem, which utilizes Discrete Cosine Transform and CUDA-based algorithm for obtaining selectivity estimation. There are also some remarks about efficiency and advantages of this approach.

Keywords: Query Selectivity Estimation, Discrete Cosine Transform, CUDA.

1 Introduction

Selectivity estimation is a process that allows a database query optimizer to estimate a query result before the query is really executed. The selectivity estimation is preformed at a very early stage of query processing. Computation of this value allows optimizer to calculate query cost and finally choose the optimal query execution plan. For a single-table query the selectivity value is the number of rows satisfying given condition divided by the number of all rows in that table. For a single-table range query with condition based on many attributes with continuous domain the selectivity may be defined as follows:

Dariusz Rafal Augustyn · Sebastian Zederowski
Silesian University of Technology, Institute of Computer Science,
16 Akademicka St., 44-100 Gliwice, Poland
e-mail: draugustyn@polsl.pl, sebastian.zederowski@gmail.com

$$\begin{aligned} & sel(Q(a_1 < X_1 < b_1 \wedge \dots \wedge a_D < X_D < b_D)) = \\ & = \int_{a_1}^{b_1} \dots \int_{a_D}^{b_D} f(x_1, \dots, x_D) dx_1 \dots dx_D \end{aligned} \quad (1)$$

where $i = 1..D$, X_i – a table attribute, a_i and b_i – query bounds, $f(x_1, \dots, x_i, \dots, x_D)$ – a probability density function (PDF) describing joint distribution of $X_1 \times \dots \times X_D$.

As we can see in eq. (1) the selectivity estimation requires estimator of multivariate PDF function. Use of multidimensional histogram as a representation of PDF is not a space-efficient solution (especially for high dimensionality). So there are many approaches to achieve a space-efficient distribution representation, such as those including: kernel estimator [9], Bayesian Network [4], Discrete Cosine Transform (DCT) [6], Cosine Series [10], Discrete Wavelet Transform [2], Clustering-based Histogram [3]. They may be implemented in Database Management System as an extension of query optimizer (e.g. the solution from [1] for Oracle DBMS optimizer). This paper affects to DCT-based method of selectivity estimation. It concentrates on methods of efficient calculation of selectivity value using GPU (Graphical Processing Unit). The paper describes the CUDA-based algorithm for selectivity estimation.

2 Selectivity Estimation Based on Discrete Cosine Transform

2.1 Representation of Attributes Values Distribution Based on DCT and DCT-Based Selectivity Estimation Method

Selectivity estimation method which uses DCT was proposed in [6]. A space-efficient DCT spectrum-based representation of distribution database table attributes values was described there.

For a 2-dimensional case, where:

X_1, X_2 – attributes of a table; both with continuous domain,

F – $\{f(i, j) : i = 0, \dots, M_1 - 1 \wedge j = 0, \dots, M_2 - 1\}$;

$M_1 \times M_2$ matrix of attributes frequencies,

a histogram estimator of PDF for joint distribution of X_1 and X_2 ,

G – $\{g(u_1, u_2) : u_1 = 0, \dots, M_1 - 1 \wedge u_2 = 0, \dots, M_2 - 1\}$,

$M_1 \times M_2$ matrix of DCT coefficients; DCT spectrum,

the 2-dimensional Discrete Cosine Transform is defined as follows:

$$\begin{aligned} g(u_1, u_2) = & \sqrt{\frac{2}{M_1}} k_{u_1} \sum_{i=0}^{M_1-1} \left\{ \sqrt{\frac{2}{M_2}} k_{u_2} \sum_{j=0}^{M_2-1} f(i, j) \cos\left(\frac{(2j+1)u_2\pi}{2M_2}\right) \right\} \cdot \\ & \cdot \cos\left(\frac{(2i+1)u_1\pi}{2M_1}\right) \end{aligned}$$

$$\text{where } k_r = \begin{cases} 1/\sqrt{2} & \text{for } r = 0 \\ 1 & \text{for } r \neq 0, \end{cases}$$

(2)

and the 2-dimensional Inverse Discrete Cosine Transform (IDCT) is defined below:

$$f(i, j) = \sqrt{\frac{2}{M_1}} \sum_{u_1=0}^{M_1-1} k_{u_1} \left\{ \sqrt{\frac{2}{M_2}} \sum_{u_2=0}^{M_2-1} k_{u_2} g(u_1, u_2) \cos\left(\frac{(2j+1)u_2\pi}{2M_2}\right) \right\} \cdot \cos\left(\frac{(2i+1)u_1\pi}{2M_1}\right). \quad (3)$$

A well-known DCT property - the energy compaction - makes possible to create a space-efficient representation of PDF, especially for high dimensionality [6]. For correlated data the spectrum coefficients g are large for small values of u_1, \dots, u_D . This means that significant coefficients are located near the origin of coordinate system in $U_1 \times \dots \times U_D$ space. Small coefficients for large values of u_1, \dots, u_D can be omitted without significant losing of representation accuracy.

The DCT-based method of query selectivity was introduced in [6]. The most important advantage of this method is that the selectivity may be calculated directly from DCT-spectrum i.e. without IDCT calculation.

The method of selectivity estimation will be shown for so-called 2-dimensional range query i.e. $Q(a_1 < X_1 < b_1 \wedge a_2 < X_2 < b_2)$.

Let us assume that $X_1 \times X_2$ space $[0, 1]^2$ is divided into $M_1 \cdot M_2$ partitions using set of pairs (x_{1i}, x_{2j}) :

$$x_{1i} = \frac{2i+1}{2M_1}, \quad x_{2j} = \frac{2j+1}{2M_2}, \quad i = 0, \dots, M_1-1, \quad j = 0, \dots, M_2-1. \quad (4)$$

Using formulas [3] and [4] we obtain $f_{X_1 X_2}(x_{1i}, x_{2j}) = f(i, j)$:

$$\begin{aligned} f_{X_1 X_2}(x_{1i}, x_{2j}) &= \\ &= \sqrt{\frac{2}{M_1}} \sum_{u_1=0}^{M_1-1} k_{u_1} \left\{ \sqrt{\frac{2}{M_2}} \sum_{u_2=0}^{M_2-1} k_{u_2} g(u_1, u_2) \cos(x_{2j} u_2 \pi) \right\} \cos(x_{1i} u_1 \pi). \end{aligned} \quad (5)$$

Thus the selectivity of Q can be obtained from:

$$\begin{aligned} sel(Q) &= \int_{a_1}^{b_1} \int_{a_2}^{b_2} f_{X_1 X_2}(x_1, x_2) dx_1 dx_2 = \int_{a_1}^{b_1} \sqrt{\frac{2}{M_1}} \sum_{u_1=0}^{M_1-1} k_{u_1} \cdot \\ &\cdot \left\{ \int_{a_2}^{b_2} \sqrt{\frac{2}{M_2}} \sum_{u_2=0}^{M_2-1} k_{u_2} g(u_1, u_2) \cos(x_2 u_2 \pi) dx_2 \right\} \cos(x_1 u_1 \pi) dx_1. \end{aligned} \quad (6)$$

Finally we obtain the selectivity value as follows:

$$\begin{aligned} sel(Q) &= \sqrt{\frac{2}{M_1}} \sqrt{\frac{2}{M_2}} k_0 k_0 g(0, 0) (b_1 - a_1) (b_2 - a_2) + \\ &+ \sum_{u_1=0}^{M_1-1} \left(k_{u_1} k_0 g(u_1, 0) \frac{\sin(u_1 \pi b_1) - \sin(u_1 \pi a_1)}{\pi u_1} (b_2 - a_2) \right) + \\ &+ \sum_{u_2=0}^{M_2-1} \left(k_0 k_{u_2} g(0, u_2) (b_1 - a_1) \frac{\sin(u_2 \pi b_2) - \sin(u_2 \pi a_2)}{\pi u_2} \right) + \\ &+ \sum_{u_1=0}^{M_1-1} \sum_{u_2=0}^{M_2-1} \left(k_{u_1} k_{u_2} g(u_1, u_2) \frac{\sin(u_1 \pi b_1) - \sin(u_1 \pi a_1)}{\pi u_1} \frac{\sin(u_2 \pi b_2) - \sin(u_2 \pi a_2)}{\pi u_2} \right). \end{aligned} \quad (7)$$

The formula [7](#) is based on full DCT spectrum. Basing on the mentioned energy compaction property we can use only significant part of the spectrum maintaining satisfactory accuracy of the estimation. This part of the spectrum is called a sampling zone and it is denoted by Z . The experimental results from [6](#) show that the reciprocal sampling zone defined as follows:

$$Z = \{(u_1, u_2) : (u_1 + 1)(u_2 + 1) \leq B\} \wedge B = \text{const} \quad (8)$$

is error-optimal for a given size of Z . Thus the estimator of selectivity of Q based on sampling zone Z can be obtained from:

$$\begin{aligned} \widehat{\text{sel}}(Q) = & \sqrt{\frac{2}{M_1}} \sqrt{\frac{2}{M_2}} k_0 k_0 g(0, 0) (b_1 - a_1) (b_2 - a_2) + \\ & + \sum_{(u_1, 0) \in Z \wedge u_1 \neq 0} \left(k_{u_1} k_0 g(u_1, 0) \frac{\sin(u_1 \pi b_1) - \sin(u_1 \pi a_1)}{\pi u_1} (b_2 - a_2) \right) + \\ & + \sum_{(0, u_2) \in Z \wedge u_2 \neq 0} \left(k_0 k_{u_2} g(0, u_2) (b_1 - a_1) \frac{\sin(u_2 \pi b_2) - \sin(u_2 \pi a_2)}{\pi u_2} \right) + \\ & + \sum_{\substack{(u_1, u_2) \in Z \wedge \\ \wedge u_1, u_2 \neq 0}} \left(k_{u_1} k_{u_2} g(u_1, u_2) \frac{\sin(u_1 \pi b_1) - \sin(u_1 \pi a_1)}{\pi u_1} \frac{\sin(u_2 \pi b_2) - \sin(u_2 \pi a_2)}{\pi u_2} \right). \end{aligned} \quad (9)$$

2.2 The Algorithm for DCT-Based Selectivity Calculation

Let us assume that a sparse matrix denoted by $zMatrix$ represents a set of spectrum coefficients $g(u_1, \dots, u_D)$ which belong to a sampling zone Z . Each element of $zMatrix$ consists of one g value and u_1, \dots, u_D values (fig [11](#)), $zMatrix.length$ is a size of the sampling zone (the value of size is significantly less than $M_1 \cdot \dots \cdot M_D$). It will be denoted by N . $zMatrix.dimension$ is dimensionality of the distribution representation and it is equal to D . $boundaries$ is a vector of query bounds – a set of pairs a_j and b_j for $j = 1 \dots D$ (fig [11](#)).

The CPU-based algorithm of selectivity calculation for a D -dimensional query written in C language is presented below:

```

01 float static CPUComputeSelectivity (SamplingZone& zMatrix,
                                     float*boundaries){
02   float el, sum = 0.0, a, b; UINT16 val;
03   unsigned int zMatrixElementSize
       = sizeof(int) + zMatrix.dimension * sizeof(UINT16);
04   const float ksqr = 1 / sqrt(2.0);
05   // for whole sampling zone
06   for(int i = 0; i < zMatrix.length; ++i){
07     // g - spectrum coefficient value
08     g = *(float*)(zMatrix.elements + zMatrixElementSize * i);
09     el = g;
10     // for every dimension
11     for(int d = 0, j = 0; j < zMatrix.dimension; ++j, d+=2) {
12       // u - index in j-dimension of spectrum
13       u = *(UINT16*)(zMatrix.elements + zMatrixElementSize * i

```

```

+ sizeof(float) + sizeof(UINT16) * j);
14 // a, b - boundaries in j-dimension of spectrum
15 a = boundaries[d]; b = boundaries[d + 1];
16 if(u != 0) {
17 // e1 == 1; ku == 1
18 e1 = e1 * (sin(u * PI * b) - sin(u * PI * a)) / PI / u;
19 } else {
20 e1 *= ksqrt2; // sqrt(2.0)
21 e1 = e1 * (b - a);}
22 } // for d, j
23 sum = sum + e1;
24 } // for j
25 return sum ; } // end of CPUComputeSelectivity function

```

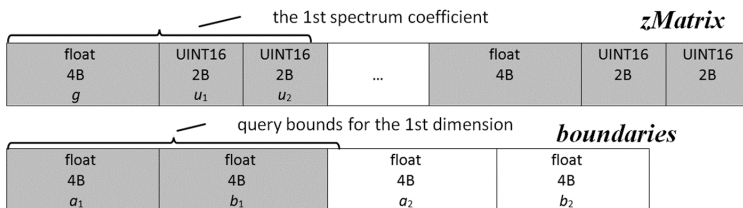


Fig. 1 Spectrum sparse matrix and vector of query boundaries representation for 2-dimensional case

The program is based on formula [9] but it is extended for any dimensions. For simplicity the above-presented program calculates the selectivity estimator but the obtained value is not multiplied by factors $\sqrt{2/M_i}$ for $i = 1 \dots D$.

The complexity of the program can be calculated from [6]:

$$2Dcost(sin)N \quad (10)$$

where $cost(sin)$ is a cost of computing *sine* function. It linearly depends on the number of coefficients in the sampling zone.

The efficiency of the program execution may be significantly improved by introducing CUDA technology, i.e. by enabling the concurrent processing of each spectrum coefficient by a separate thread in GPU.

3 CUDA-Based Selectivity Estimation Routine

CUDA technology [7] may enable fast creation of DCT spectrum as a representation of multidimensional attributes values distribution. For example, this may be achieved by using CUFFT library [8]. But more important for on-line query processing is efficient calculating selectivity values by DBMS

cost optimizer. It is time critical operation. Because we assume whole time of a query processing as about one second, so we also assume that a time of selectivity calculations should be significantly less i.e. it should be no greater than one millisecond. We propose the solution which satisfies this. It is based on CUDA technology and utilizes existing DCT spectrum.

In the proposed algorithm, we assume that each coefficient from $zMatrix$ is processed by one thread. This is realized by the execution of a so-called kernel routine at GPU side.

Let us assume that $zMatrix$ containing coefficients from a reciprocal sampling zone (spectrum matrix content) are loaded into global memory of CUDA device. This can be done during the DBMS initialization. Let ${}_d_matrix$ denote the handle of spectrum at the GPU side. For every processed query the DBMS optimizer has to load its *boundaries* into CUDA device. Let ${}_d_boundaries$ denote the handle of *boundaries* at the GPU side.

CUDA usage requires to set the size of thread block and the size of the data grid [7]. The block size denoted by $block_x$ is equal to 64. The grid size depends on $zMatrix$ size. It is denoted by $grid_x$ and is equal to $\lceil zMatrix.length/block_x \rceil$.

Initially either the spectrum ($zMatrix$) or the query bounds (*boundaries*) are loaded into the global GPU memory. Because each thread uses every query bound, the *boundaries* vector is copied from the global memory into the fast shared memory by the kernel routine. The size of allocated shared memory is denoted by $shMemSize$ and it is initially set to the size of *boundaries*.

The kernel routine may be invoked as follows:

```
dim3 grid(grid_x, 1, 1); dim3 threads(block_x, 1, 1);
GPUComputeSelectivity <<<grid, threads, shMemSize>>> (_d_matrix,
    zMatrix.dimension, zMatrix.length, _d_boundaries, _d_part_res);
```

Finally the result selectivity is obtained by summing all elements from ${}_d_part_res$ (after copying them into host memory). This is done at CPU side.

At the beginning of the kernel source code (*GPUComputeSelectivity* function) the shared memory variables for placing boundaries and partial results are declared as follows:

```
01 __global__ void GPUComputeSelectivity (BYTE* zmatrix, int dim,
    int len, float* bound, float* part_res) {
02 __shared__ float pRes[BLOCK_SIZE]; // sh. mem. for part.results
03 extern __shared__ float s_bound[]; // shared memory for bounds
```

Predefined built-in variables [7]: $threadIdx.x$, $blockIdx.x$, $blockDim.x$ allow to set: the index of the thread in the currently processed block (tid), the index of the currently processed matrix element ($tIdx$) and the location of the current matrix element (pos), i.e. the address of currently processed spectrum coefficient value:

```
04 const unsigned short int tid = threadIdx.x;
05 const unsigned short int tIdx = blockIdx.x * blockDim.x + tid;
06 const BYTE* pos = zmatrix + size(float)
    + sizeof(UINT16) * dim * tIdx;
```

A few threads (*dim* threads) are used for copying *boundaries* into shared memory (into *s_bound*):

```

07  if(tid < dim) {
08      s_bound[tid] = bound[tid];
09      s_bound[tid + dim] = bound[tid + dim];
10  }
11  __syncthreads(); // Wait for end of all copying

```

The source code of the kernel main processing - the calculation for single coefficient for all dimensions - is presented below. This is very similar to source code lines 11 ~ 22 of *CPUComputeSelectivity*.

```

12  // Read current matrix element e1 - single spectrum coefficient
13  float e1 = *(float*)(pos);  UINT16 u;
14  // Seek position to first dimension index for current element
15  pos += sizeof(float);
16  for(int d = 0, i = 0; i < dim; ++i, d += 2) {
17      // Get the dimension index
18      u = *(UINT16*)(pos + sizeof(UINT16) * i);
19      if(val != 0){
20          e1 = e1 * (__sinf(u * PI * s_bound[d + 1])
21                  - __sinf(u * PI * s_bound[d])) / PI / u;
22      } else {
23          e1 *= ksqr2; // sqrt(2.0)
24          e1 = e1 * (s_bound[d + 1] - s_bound[d]);
25      }
26  } // for
27  // Store result into shared memory
28  pRes[tid] = e1;
29  __syncthreads();

```

pRes values are summed by the first thread (*tid* = 0) in current block so *part_res* vector contains sums from all blocks. This is shown below:

```

30  if (tid == 0){ float tmp = 0.0 ;
31      for(int i = 0; i < blockDim.x; ++i)
32          tmp += pRes[i];
33      part_res[blockIdx.x] = tmp;
34  }
35 } // end of GPUComputeSelectivity kernel

```

part_res is copied to host into *_d_part_res*. Elements of *_d_part_res* array are summed by CPU (what was already mentioned).

The above-described basic version of the algorithm was improved by introducing some CUDA-based mechanisms, i.e.:

- utilizing fast texture memory,
- summing the partial results inside a block by several threads (using reduction mechanism [5]),
- invoking additional (the second) kernel for calculating final sum (which eliminates usage of CPU).

Those three extensions were introduced in the final version of the algorithm described below.

The texture memory supports a cache mechanism. Threads of the same warp that read texture addresses that are close together achieves best performance [7]. For that reason *zMatrix* content was enabled to the new kernel by the texture memory (therefore *zMatrix* is not a parameter of the kernel function anymore).

Instructions are SIMD synchronous within a warp [7, 5]. Warp is a set of 32 threads that are executed in parallel and synchronously. This allows to enroll the loop described by lines 31 ~ 32 of the basic kernel and involve 32 threads to calculate the sum concurrently. This may improve efficiency, because previously only one thread was used for summing in the basic kernel version. It is possible to apply due to a block size set to 64. Lines 30 ~ 34 of the basic kernel are replaced by the source code listed below:

```

01  if (tid < 32){
02  // Warp-synchronization needs to declare shared memory volatile.
03  volatile float* smem = pRes;
04  pRes[tid] = e1 = e1 + pRes[tid + 32];
    . . .
05  pRes[tid] = e1 = e1 + pRes[tid + 2];
06  pRes[tid] = e1 = e1 + pRes[tid + 1];
07  }
08  if (tid == 0) // Write result for this block to global memory
09  part_res [blockIdx.x] = pRes[0];

```

Partial results (*_d_part_res* vector) i.e. a sums of processed spectrum coefficients for each block are placed in the global memory. This is done by the above-described kernel (either the basic or the final one). The additional kernel – designated only for calculating the one final value of selectivity i.e. sum of partial results – was also developed.

4 Experimental Results

Experimental verification was made using a low-budget GPU device GeForce 8600M GT and CPU Intel Core 2 Duo T8300 2.40 GHz. This graphic card has 4 multiprocessor, 8 cores per multiprocessor, 256MB total memory and 16kB shared memory per block.

We assumed that DCT spectrum coefficients (*zMatrix*) had been already loaded into GPU memory. Obtained times of execution relates to a selectivity calculation for a given query bounds. Experiments were performed for 2 ~ 49 dimensions and for selected sample sizes (set of N – numbers of spectrum coefficients in *zMatrix*) that belong to 512, 1024, 2048, 4096. Both versions of the algorithm (i.e. basic and final one) were considered.

Fig. 2a presents execution times of CPU program and GPU one (the basic algorithm) where $N = 512$. CPU time almost linearly depends on D (see eq. 10) and its values changes from 0.1 to 2.38 ms. GPU times are significantly smaller. GPU time is almost constant. For $N = 4096$ and $D = 49$ CPU-based program execution takes about 19.3 ms (the worst case). GPU-based solution

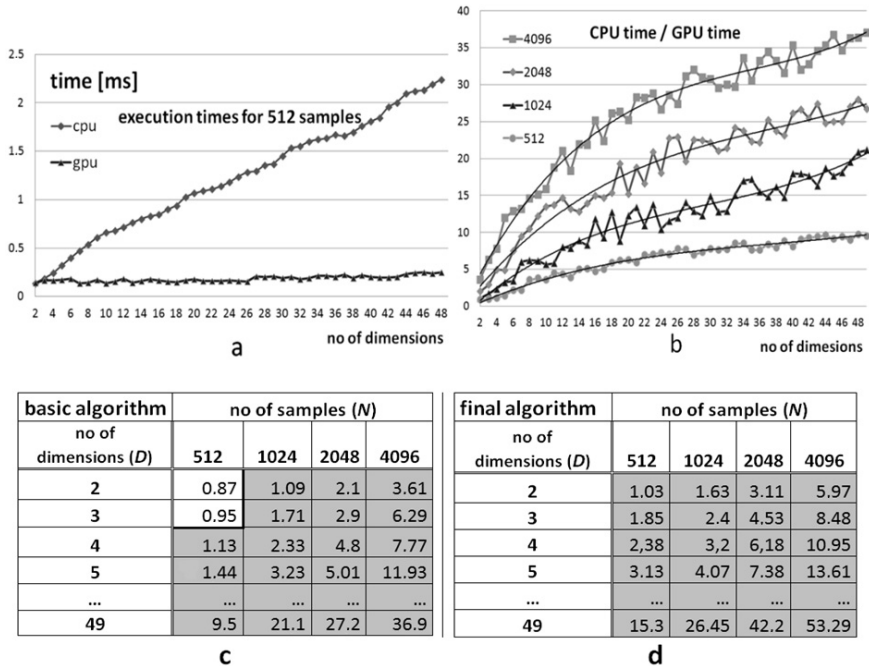


Fig. 2 a) Execution times for CPU and GPU for $N = 512$ (basic algorithm), b) GPU program speedup relative to CPU one for $N = 512, 1024, 2048, 4096$ (basic algorithm), c) Detailed values of GPU program speedup relative to CPU one for the basic algorithm d) Speedup for the final algorithm

(the final algorithm) is faster about 53.3 times (fig. 2.d) so the execution time is equal to 0.36 ms which satisfies the previously mentioned assumed condition that a selectivity calculation should be rather less than 1 ms.

Fig. 2.b presents GPU program speedup relative to CPU one (values of CPU execution time divided by GPU one). A one sample instance of experimental results and mean results are shown for each value of N . Fig. 2.b shows the nearly linear dependency between speedup and number of dimensions. Speedups becomes larger for high values of N . Detailed values of speedup were shown in fig. 2.c. For $N = 512$ applying GPU is valuable (the speedup value is greater than 1) when $D > 3$.

Experiments also included the use of the program implementing the final algorithm. Detailed data about the GPU program speedup related to CPU one is presented in fig. 2.d. Comparing fig. 2.c and fig. 2.d we can see that results are much better for the efficient final algorithm. In particular, the GPU applying is the beneficial for all values of D and N .

5 Conclusions

The paper affects the problem of query selectivity estimation for queries with complex conditions based on many table attributes. Known unconventional method based on Discrete Cosine Transform is considered.

The CUDA-based algorithm and its efficient implementation were proposed in the paper. Selected GPU mechanisms applied for speeding up the DCT-based selectivity estimation were described. Advantages of the proposed solution (comparing to CPU-based one) were shown.

CUDA-based approach may be also useful for applying in other selectivity estimation methods mentioned in introduction section, e.g. in wavelet-based one.

References

1. Augustyn, D.R.: Applying Advanced Methods of Query Selectivity Estimation in Oracle DBMS. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) *Man-Machine Interactions*. AISC, vol. 59, pp. 585–593. Springer, Heidelberg (2009)
2. Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K.: Approximate query processing using wavelets. *The VLDB Journal* 10, 199–223 (2001)
3. Furfaro, F., Mazzeo, G.M., Sirangelo, C.: Exploiting Cluster Analysis for Constructing Multi-dimensional Histograms on Both Static and Evolving Data. 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. 442–459. Springer, Heidelberg (2006)
4. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. *SIGMOD Rec.* 30, 461–472 (2001)
5. Harris, M.: *Optimizing Parallel Reduction in CUDA* (2011), http://www.uni-graz.at/~haasegu/Lectures/GPU_CUDA/Lit/reduction.pdf
6. Lee, J.H., Kim, D.H., Chung, C.W.: Multi-dimensional selectivity estimation using compressed histogram information. In: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD 1999, pp. 205–214. ACM, New York (1999), <http://doi.acm.org/10.1145/304182.304200>, doi:10.1145/304182.304200
7. NVidia Corporation: *NVIDIA CUDATMC Programming Guide*, version 4.1 (2011), <http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA.C.Programming.Guide.pdf>
8. NVidia Corporation: *NVIDIA GPU Computing Documentation*, *CUFFT Library User Guide* (2012), <http://developer.nvidia.com/nvidia-gpu-computing-documentation>
9. Scott, D.W., Sain, S.R.: *Multi-Dimensional Density Estimation*, pp. 229–263. Elsevier, Amsterdam (2004)
10. Yan, F., Hou, W.C., Jiang, Z., Luo, C., Zhu, Q.: Selectivity estimation of range queries based on data density approximation via cosine series. *Data Knowl. Eng.* 63, 855–878 (2007)

Processing of Range Query Using SIMD and GPU*

Pavel Bednář, Petr Gajdoš, Michal Krátký, and Peter Chovanec

Abstract. Onedimensional or multidimensional range query is one of the most important query of physical implementation of DBMS. The number of compared items (of a data structure) can be enormous especially for lower selectivity of the range query. The number of compare operations increases for more complex items (or tuples) with the longer length, e.g. words stored in a B-tree. Due to the possibly high number of compare operations executed during the range query processing, we can take into account hardware devices providing a parallel task computation like CPU's SIMD or GPU. In this paper, we show the performance and scalability of sequential, index, CPU's SIMD, and GPU variants of the range query algorithm. These results make possible a future integration of these computation devices into a DBMS kernel.

1 Introduction

Range query (or range scan) [19] is one of the most important query of physical implementation of DBMS [16]. There are two basic variants: onedimensional and multidimensional range query. DBMS often utilize two types of data structures (and algorithms) supporting these range queries: a sequential scan in a persistent array (called the heap table) of blocks (4, 8, or 16 kB in size) or a range scan in a data structure (mostly a tree, called the index). Although it seems that the sequential scan is very inefficient, it is not generally true; there are situations where a DBMS processes the range scan in a sequence array instead the range query in a tree. It is because the range query processing in a tree is influenced by random accesses in the

Pavel Bednář · Petr Gajdoš · Michal Krátký · Peter Chovanec
Department of Computer Science, VŠB – Technical University of Ostrava, Czech Republic
e-mail: [pavel.bednar, petr.gajdos, michal.kratky}@vsb.cz](mailto:{pavel.bednar, petr.gajdos, michal.kratky}@vsb.cz),
peter.chovanec@vsb.cz

* Work is partially supported by Grant of GACR No. GAP202/10/0573.

secondary storage as well as the main memory [15, 6]. As result, particularly in the case of a high number of accessed blocks (or nodes or pages) in an index, the query processing time can be lower compared to an index even though the sequential scan compares more items than the index.

Onedimensional range query is often implemented in a data structure like a B-tree [1] and it can be processed in an execution plan of the following SQL statement: `SELECT * FROM T WHERE $q_{l_1} \leq T.attr_1 \leq q_{h_1}$` . Multidimensional range query is often implemented by multidimensional data structures, e.g. n -dimensional B-tree [8], R-tree [10] or R*-tree [2], and it can be processed for the following SQL statement: `SELECT * FROM T WHERE $q_{l_1} \leq T.attr_1 \leq q_{h_1}$ AND ... AND $q_{l_n} \leq T.attr_n \leq q_{h_n}$` . In other words, this query retrieves all tuples of an n -dimensional space matched by an n -dimensional query rectangle.

Although the point query can be considered as a special type of the range query, an important difference is in the possibly high number of nodes accessed during the range query processing over an index. The number of compared items (of an index) can be enormous especially for lower query selectivity where the result includes more items. The number of compare operations increases for more complex items (or tuples) with the longer length, e.g. words stored in a B-tree. In the case of the multidimensional range query, the longer length means the higher dimension of a multidimensional space.

Due to the possibly high number of compare operations executed during the range query processing, we can take into account hardware devices providing a parallel task computation like CPU's SIMD or GPU [11]. There are some works related to the utilization of SIMD and GPU in DBMS. In [4], authors introduced an efficient implementation of sorting on a multi-core CPU SIMD architecture. A full table scan using SIMD has been shown in [24]. In [25], authors introduced a utilization of SIMD in some database operations. In [3], authors depict some preliminary works of an integration of these hardware devices into a DBMS kernel. In [9], GPU is utilized for metric searching. In our previous work [5], we compared the CPU's SIMD range query algorithms for the R-tree and sequential array. We found out that the SIMD variant improves the range query processing at most $2\times$. In our best knowledge, there is no work comparing sequential, index, CPU's SIMD, and GPU variants of the range query algorithm. Since it is necessary to know the performance and scalability of these algorithms for their correct integration into a DBMS kernel, we introduce the comparison in this article.

In generally, there are no significant differences between one and multi-dimensional range queries; both range queries must compare individual values of a tuple. We aim our effort to the multidimensional range query in this article. The outline of the paper is as follows. Section 2 briefly describes CPU's SIMD and GPU technologies. Section 3 presents CPU's SIMD and GPU implementations of the range query algorithm. In Section 4, we put forward experimental results. These results show some limits of hardware devices used; therefore, we conclude with a discussion and outline possible ways of our future work.

2 Overview of CPU's SIMD and GPU

2.1 CPU's SIMD and SSE

SIMD (Single Instruction, Multiple Data) instructions can increase the performance when exactly the same operations are performed on multiple data objects [11]. Typical applications are communication, digital signal, and graphics processing [20, 21]. Various CPU systems based on the SIMD paradigm have been introduced in last decades. Systems like MMX, SSE¹, 3DNow!², AltiVec³ may be considered as the major ones (for a complete list of SIMD architectures see [22]).

Streaming SIMD Extensions (SSE) is an Intel's SIMD extension of the x86 instruction set introduced in 1999; it was subsequently expanded by Intel to the current version SSE5. SSE includes scalar and packed instructions over eight 128 bit-length registers xmm0 – xmm7. The xmm register can include 16–2 integers (of 8–64 bit-lengths) and 4–2 floating-point numbers (single or double precision). In this way, it enables to perform, for example, 4 parallel operations over 32 bit-length integers or floats. 3DNow! provides 64 bit-length registers; however, AltiVec provides the same functionality as SSE. Intel's AVX⁴ supports 256 bit-length registers (currently reduced to only floating-point numbers) and thus 8 parallel operations over 32-bit-length data types. SSE contains over 200 scalar or packed instructions (e.g. arithmetic, comparisons and so on). In our SIMD range query implementation (see Section 3.2), packed compare instructions are used (see Figure 1).

2.2 GPU and Cuda

Architecture of GPUs (Graphics Processing Units) is suitable for vector and matrix algebra operations. That leads to the wide usage of GPUs in the area of information retrieval, data mining, image processing, data compression, etc. [13]. There are two graphics hardware vendors: ATI and nVIDIA. ATI develops technology called ATI Stream⁵ and nVIDIA presents nVIDIA CUDA⁶. Nowadays, programmers usually choose between OpenCL which is supported by ATI and nVIDIA [12], and CUDA which is supported by nVIDIA only [13]. An important benefit of OpenCL is its

¹ <http://software.intel.com/en-us/articles/using-intel-streaming-simd-extensions-and-intel-integrated-performance-primitives-to-accelerate-algorithms/>

² <http://www.amd.com/us/products/technologies/3dnow/Pages/3dnow.aspx>

³ <http://www.freescale.com/webapp/sps/site/overview.jsp?code=DRPPCALTVC>

⁴ <http://software.intel.com/en-us/articles/introduction-to-intel-advanced-vector-extensions>

⁵ <http://www.amd.com/stream/>

⁶ http://www.nvidia.com/object/cuda_home_new.html

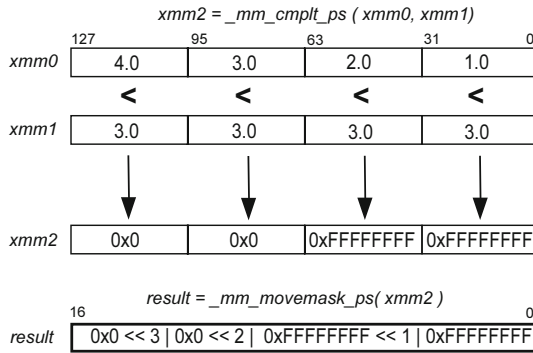


Fig. 1 SSE packed comparison of 4 float values

platform independence; however, CUDA still sets the trends in GPU programming. This article is not focused on a detail comparison of these two approaches; we utilize CUDA in our experiments.

CUDA (Compute Unified Device Architecture) is a general purpose parallel computing architecture. GPUs utilized in our experiments are based on the Fermi architecture [18] which is still the most common GPU architecture since the original G80. Currently, the new architecture called Kepler has been introduced by nVIDIA.

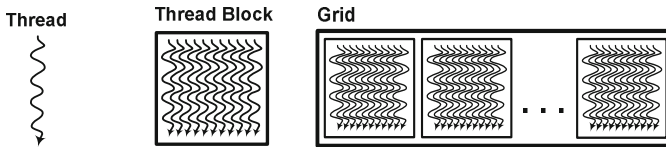


Fig. 2 A schema of the CUDA threads arrangement [18]

GPUs of the Fermi architecture include a number of Streaming Multiprocessor (SM) with 32 cores, e.g. nVIDIA Tesla 2050 provides 14 SM with 448 CUDA cores. A CUDA program calls parallel kernels. A kernel executes in parallel across a set of parallel threads. The programmer or compiler organizes these threads in thread blocks and grids of the thread blocks. Each thread within a thread block executes an instance of the kernel. The GPU instantiates a kernel program on a grid of parallel thread blocks. The simplified arrangement of threads is illustrated in Figure 2. A thread block is a set of concurrently executing threads that can cooperate among themselves through a barrier synchronization and shared memory. A grid is an array of thread blocks that execute the same kernel, read inputs from global memory, write results to global memory, and synchronize between dependent kernel calls. For more detail we refer to [18].

3 Multidimensional Range Query Processing

3.1 Introduction

In the case of the multidimensional range query, we have two primitive operations: *IsInRectangle*, returning true if a tuple is in the query rectangle, and *IsIntersected*, returning true if a rectangle (or MBR – minimal bounding rectangle) intersects the query rectangle. These operations are used in both sequential as well R-tree range query implementations. Since, in our experiments, we test only point data, we invoke only the *IsInRectangle* operations for the sequence scan. Both these operations must perform two compare operations for each dimension. However, in [5], we show that it is not necessary to check all dimensions in the case of the conventional implementation and it is the reason why the SIMD implementation is not always successful.

3.2 Range Query Processing Using CPU's SIMD

In [5], we depicted the SIMD variants of operations *IsIntersected* and *IsInRectangle*. These SIMD implementations work correctly only if the space dimension n is a multiple of *PackCount* (the number of values fit in one SIMD register, *PackCount* = 4 for SSE and 32 bit-length values). In Algorithm 1, we introduce the new SIMD *IsInRectangle* algorithm handling tuples of any space dimension. The function *GetResult* returns the result of a packed comparison depending on the space dimension. The new *IsIntersected* then includes analogous improvements.

We proposed some additional optimizations. Loading the query rectangle into the SIMD registers can be excluded from each function invocation when $n \leq \text{PackCount}$ (when we do not store the query rectangle in more than two registers); the values are loaded at the beginning of the range query. Moreover, these functions can process multiple tuples at once. For example, if *PackCount* = 4 and $n = 2$ we can compare two couples of tuples at once. This optimization is more usable in the case of AVX with 8×32 bit-length registers, where we are able to compare up-to four couples of 2D, three 3D or two 4D tuples at once. This technique causes some changes in Algorithm 1, since *IsInRectangle* returns more than one result.

Although the SIMD algorithms cause less branch misspredictions than conventional operations [11], in [5], we show that it improves the performance at most $2 \times$. In Section 4, we put forward that our new implementation is only little bit faster than the original SIMD algorithms. However, the improvement of the new implementation is up-to $6 \times$ when we utilize the proposed optimizations and AVX.

Algorithm 1: New SIMD IsInRectangle Algorithm

```

Input : Integer array pql (QL) and Integer array pqh (QH), Tuple T, Space dimension  $n$ 
Output: true if T is in QR (defined by QL and QH), otherwise false

1 if ( $n \% \text{PackCount} > 0$ ) then
2   cycles = ( $n / \text{PackCount}$ ) + 1;
3 else
4   cycles =  $n / \text{PackCount}$ ;
5 end
6 remainingCompares =  $n$ ;
7 for  $i \leftarrow 0$  to cycles do
8   if ( $\text{remainingCompares} \geq \text{PackCount}$ ) then
9     remainingCompares = remainingCompares - PackCount; compares = PackCount;
10  else
11    compares = remainingCompares
12  end
13  QRQL = _mm_load_ps(pql[i * PackCount]);
14  QRQH = _mm_load_ps(pqh[i * PackCount]);
15  reg = _mm_load_ps(T[i * PackCount]);
16  resultReg = _mm_cmplt_ps(reg, QRQL[i]);
17  if (GetResult(resultReg, compares) != 0) then
18    return false;
19
20  resultReg = _mm_cmpgt_ps(reg, QRQH[i]);
21  if (GetResult(resultReg, compares) != 0) then
22    return false;
23
24 end
25 return true;

```

3.3 Range Query Processing Using GPU

Our GPU range query algorithm (see Algorithm 2) is written in C++ and CUDA SDK for the compute capability 2.0 and higher [17]. In the area of GPU algorithms, a common technique is to arrange data for needs of the GPU algorithm. Since we suppose common row-oriented DBMS, input data are unchanged in the form of n -tuples. We do not consider another tuple arrangement, e.g. column-oriented [23].

The algorithm includes more variables, they generally differ in the first character (g, c, s, r) which indicates the type of the memory used: g = global memory, c = constant memory, s = shared memory, r = registers. Every memory type has its pros and cons (see [17] [13] for more detail). The qI represents an array of input tuples, gNI is the number of input tuples. The qNC variable defines a number of data blocks (chunks) that will be processed by a single CUDA thread block. The reason of this is in the fact that CUDA sets a limit for the number of threads in a block and kernel function. Moreover, it is useful to design CUDA kernels such that every block of threads processes a sequence of data chunks to hide memory latency and increase the performance [7].

The gQR represents a query rectangle and $qOut$ is the output boolean vector. The cD variable (stored in constant memory) includes the space dimension. The number of registers is defined by CUDA and a general rule is: the more used registers the less threads per block [13]; therefore, storing common constant data in constant memory, with respect to the kernel design, can save per thread registers and

subsequently increase the number of threads in thread blocks. The sQR is a copy of gQR . It is stored in per-block shared memory because of better performance; the shared memory is faster than global memory. The $sQRR$ is a temporary vector for tuples of a currently processed chunk, and it is stored in the per-block shared memory as well. All variables starting with r are auxiliary per-thread variables stored in registers. The algorithm has several important phases marked by ranges of lines in the code:

- [5–8] A few threads (fulfilling the condition $rTID < cD$) copy the query rectangle in shared memory to be fast accessible by all threads in the block. A synchronization point must be performed to ensure that all threads of each block will continue after the query rectangle is prepared in shared memory.
- [9] Every thread block processes a number of data chunks (gNC).
- [12–25] This part represents an *Element-Per-Thread* processing phase of a range query. For example: Let current data chunk consists of 3 input tuples ($rNIC = 3$) of dimension 8 ($cD = 8$), which is 24 values. Let the thread block has 32 thread (THREAD_PER_BLOCK constant). Consequently, first 3×8 threads test all tuples in the data chunk; threads 0, 8, 16 test the first dimension, threads 1, 9, 17 test the second dimension and so on. Finally, all 24 individual threads' results are stored in the per-block shared memory $sQRR$. All memory accesses satisfied coalesced access pattern to achieve the maximum performance [13]. This part is also closed by a synchronization point to ensure that all parallel threads finished and saved their resulting data.
- [26–33] The last part of the proposed algorithm is responsible for the final summarization and storing data into the final output vector $gOut$. When we consider the example depicted in the previous phase, all 24 ($= 3 \times 8 = rNIC \times cD$) individual results are stored in shared memory. All threads in a thread block fulfilling $rTID < rNIC$ further summarize the data in shared memory ($sQRR$) such that every thread starts with an appropriate offset and summarize 8 consequent elements. Finally, these threads store their final results into $gOut$. This phase can be extended by a parallel reduction process; however, this is inefficient in the case of a low dimension [13] [7].

There are many ways how to implement the range query algorithm using GPU. In Section 4, we show that our algorithm is up-to $8 \times$ faster than the conventional sequential algorithm. In the experiments, we show that the main issue of the GPU algorithm is the high data transfer time from the main memory to the GPU's memory. This transfer time is independent on the GPU range query algorithm used.

4 Experimental Results

In our experiments, we compare the performance of range query processing using sequential and index algorithms. In the case of the sequential scan, we compare three variants: conventional, CPU's SIMD (SSE), and GPU. In the case of

Algorithm 2: CUDA Range Query Algorithm

Input : Inputs (Tuples) gI , Number of Inputs gNI , Number of Chunks gNC , Query Rectangle gQR
Output: Output Result Set $gOut$

```

1  rTID = threadIdx.x;
2  rCFI = blockIdx.x * THREADS_PER_BLOCK * gNC;
3  rCDO = rCFI * cD;
4  rTDO = rTID;
5  if rTID < cD then
6      sQR [rTID] = gQR [rTID];
7  end
8  __syncthreads();
9  for c ← 0 to gNC do
10     rNIC = min(THREADS_PER_BLOCK, gNI-rCFI);
11     rTDO = rTID;
12     #pragma unroll 8
13     for i ← 0 to cD do
14         if rTDO < (rNIC * cD) then
15             rTQO = rTDO * cD;
16             if NOTINTERVAL(sQR [rTQO ],min, sQR [rTQO ],max, gI [rCDO +rTDO ]) then
17                 sQRR [rTDO] = false;
18             end
19             else
20                 sQRR [rTDO] = true;
21             end
22             rTDO += THREADS_PER_BLOCK;
23         end
24     end
25     __syncthreads();
26     rTDO = rTID * cD;
27     rTR = true;
28     if rTID < rNIC then
29         #pragma unroll 8
30         for i ← rTDO to (rTDO + cD) do
31             rTR &= sQRR [i];
32         end
33         gOut [rCFI + rTID] = rTR;
34     end
35     rCFI += rNIC;
36     rCDO = rCFI * cD;
37 end

```

the R-tree, we compare two variants: conventional and SSE. All SSE experiments have been executed with aligned memory access without software prefetching since data structures are organized as blocks to be scanned without random accesses. All data structures have been implemented in C++ and compiled for x86 a x64 by Microsoft C++⁷. The difference between both platforms is relatively low (approximately 10%); therefore, we ignore it. Since we want to compare only the main-memory run in all cases; we ignore disk I/O costs.

In our test, we utilized 5 real collections. The first collection, titled XML, represents a set of paths in the XMark⁸ collection [14]. The second collection, titled CARS⁹, includes spatial records related to California, USA. The third collection

⁷ <http://msdn.microsoft.com/visualc>

⁸ <http://monetdb.cwi.nl/xml/>

⁹ <http://www.census.gov/geo/www/tiger/>

is TIGER^[10], that is a standard Tiger/Line data set for testing spatial databases. We chosen the Wyoming data set from 2006 and index type 2 and have not been consider any topological information. The fourth collection WORDS^[11] contains tuples from text collections. The fifth collection STOCKS^[12] represents historical stock data from 1970–2010. In Table 1 we see characteristics of these data collections. In our experiments^[13], we tested the performance for 40 range queries for each data collection divided to 4 query groups according to the selectivity. All range queries were 10× repeatedly executed and results have been averaged for one query. The time of adding tuples into a result set has not been measured, since it is the same for all algorithms.

Table 1 Test Data Collections

Collections (Dimension)	XML (8)	CARS (4)	TIGER (2)	WORDS (3)	STOCK (11)
#Tuples	15,884,160	3,318,583	5,889,786	483,450,157	19,610,499
Domain	Integer	Float	Integer	Integer	Float
Tuple Size [B]	32	16	8	12	44
Size [MB]	349	106	112	7,720	997

In [5], we showed that successfulness of the SSE implementation depends on the number of cycles which executes the conventional *IsInRectangle* or *IsIntersected* algorithms; we call it the number of cycles (or #Cycles, see the penultimate column of Table 2). If #Cycles is close to the space dimension (it arises especially in the case of the low selectivity), we obtain at most a 2× improvement of the query processing time (see Table 2). There are two exceptions. In the case of TIGER and WORDS, a query rectangle is loaded into the xmm registers before the whole query is executed. Moreover, in the case of TIGER, we process two tuples by one packed instruction. Together with other technical improvements of our new algorithms, we obtain up-to 4× lower processing time. In the case of AVX, we can process more tuples at once compared to SSE. As result, in Table 3, we see up-to the 6× improvement compared to the conventional algorithm. Moreover, the minimal improvement is 5×, whereas in Table 2, we see that the SIMD implementation can be slower in case we can not apply any optimization.

In Table 2, we see that the R-tree is particularly successful in the case of low-selectivity queries, on the other hand, results of our GPU algorithm are approximately the same for all query selectivities. However, we must keep in mind that the R-tree processes less operations than the sequence-based approaches. Results of the R-tree with SSE are not depicted for deficit of space, we state that the query processing time of the R-tree with SSE is at most 2× lower compared to the R-tree

¹⁰ <http://www.census.gov/geo/www/tiger/>

¹¹ <http://archive.ics.uci.edu/ml/datasets/Bag+of+Words>

¹² <http://www.infochimps.com/datasets/nasdaq-exchange-daily-1970-2010-open-close-high-low-and-volume>

¹³ The experiments were executed on an Intel Core i5-P2450 3.2Ghz, 6.0 MB L2 cache; 8GB of DDR3; Windows 7 SP1.

Table 2 Results of various range query implementations

Collection (Dimension)	Query Group	Result Size	Time [s]				IsInRectangle Avg. #Cycles	IsInRectangle Calls
			Conventional	SSE	GPU	R-tree		
XML (8)	1	1.0	0.128	0.173	0.081	0.0000	1.058	15,884,160
	2	591.4	0.155	0.191	0.081	0.0002	1.374	
	3	38,668.4	0.204	0.200	0.081	0.0130	2.005	
	4	417,003.0	0.415	0.228	0.081	0.1046	4.718	
TIGER (2)	1	1.0	0.066	0.014	0.010	0.0000	1.000	5,889,786
	2	510.6	0.065	0.014	0.010	0.0009	1.002	
	3	40,992.1	0.067	0.014	0.010	0.0021	1.079	
	4	856,908.4	0.080	0.017	0.010	0.0291	1.445	
WORDS (3)	1.	1.0	6.562	2.271	1.075	0.0024	1.000	483,450,157
	2.	528.1	3.970	2.265	1.064	0.0804	1.114	
	3.	23,922.5	4.814	2.279	1.077	0.1794	1.429	
	4.	475,487.9	5.558	2.274	1.071	0.3384	1.709	
STOCKS (11)	1.	1.0	0.199	0.216	0.086	0.0001	1.414	19,610,499
	2.	519.6	0.206	0.215	0.086	0.0028	1.500	
	3.	32,538.1	0.264	0.231	0.086	0.0355	2.105	
	4.	977,986.0	0.477	0.293	0.086	1.0235	3.822	

Table 3 Query Statistics for float AVX implementation

Collection (Dimension)	Query Group	Result Size	Time [s]		IsInRectangle Avg. Cycle	IsInRectangle Calls
			Conventional	AVX		
CARS (4)	1	1.0	0.0314	0.0064	1.00008	3,360,277
	2	484.0	0.0326	0.0066	1.02825	
	3	62,077.9	0.0371	0.0064	1.25278	
	4	1,159,228.0	0.0541	0.0075	2.25448	

without SSE. The GPU variant¹⁴ is up-to $8\times$ more efficient than the conventional sequential algorithm; however, this table includes only the kernel processing time with the maximal buffer size.

Table 4 Query Statistics for GPU

Collection (Dimension)	Buffer Size [B]	Query Group	Result Size [s]	Time [s]			Total Time [s]
				Kernel	H->D	D->H	
WORDS (3)	8,168	1	1.0	212.878	23.134	16.366	252.378
		2	528.1	212.731	19.623	16.165	248.519
		3	23,922.5	212.900	19.459	16.503	248.862
		4	475,487.9	212.871	24.067	16.746	253.684
	635,278,860	1	1.0	1.075	0.870	0.075	2.019
		2	528.1	1.064	0.870	0.075	2.008
		3	23,922.5	1.077	0.869	0.074	2.020
		4	475,487.9	1.071	0.869	0.074	2.014

In Table 4, we put forward detail results of GPU over the WORDS data collection. In this table, we show results for various buffer sizes, which is the size of data transferred and searched on a GPU in one step. The kernel column includes only the

¹⁴ The experiments were executed on nVidia GeForce 550Ti with 1GB of DDR5 (Memory speed 4,104 Mhz), 4 SM, 48 cores/SM.

query processing time. H->D and D->H (H means host, D means device) include the transfer time of data on a GPU and the transfer time of an output from the GPU. Lower buffer sizes correspond to a naive integration of GPU in a DBMS kernel: if we need a range query computation over a block (mostly 8 kB in size), this block is transferred on a GPU and the range query is executed. As result, we obtain all processing times very long. In the case of the large buffer size, the time of computation is the same as the data transfer time.

5 Conclusions

In this article, we compared the conventional, SSE, and GPU variants of range query algorithms. Conventional and SSE implementations are utilized in the case of sequential as well as the R-tree variants. We summary the results. (1) For a lower space dimension, when we can use pre-loading of the query rectangle and process more tuples at once, an improvement of the sequential SSE variant is up-to $6\times$. On the other hand, if it is not possible to utilize these optimizations, the improvement is relatively low. (2) The results of GPU show the fast range query computation (up-to $8\times$ faster than the conventional sequential algorithm). However, GPU does not provide any significant improvement when we consider the data transfer time. In contrast, this issue is not, for example, reported by articles related to using GPU to metric searching [9], the reason is that a distance computation is much more expensive than the cost of both primitive range query operations. Moreover, other GPU devices must be tested, since the improvement of the GPU algorithm compared to the sequential SSE algorithm is only up-to $3\times$. (3) Due to the GPU performance, the GPU algorithms can be used in indices, e.g. in the R-tree. However, our experiments with sequence-based approaches show that we can not use a naive method with a transfer and searching of single pages (see results with the 8kB buffer size).

As result, an integration into a DBMS kernel is not so straightforward and we must solve mainly these issues in our future work: the data transfer time must be effaced in the live cycle of DBMS and we must forward only low-selectivity range queries on a GPU because the number of compare operations increases for indices like B-tree or R-tree in this case.

References

1. Bayer, R., McCreight, E.: Organization and Maintenance of Large Ordered Indexes. *Acta Informatica* 3(1), 173–189 (1972)
2. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. In: *Proceedings of the ACM International Conference on Management of Data, SIGMOD 1990* (1990)

3. Beier, F., Kiliyas, T., Sattler, K.U.: GiST Scan Acceleration using Coprocessors. In: Proceedings of 8th Int. Workshop on Data Management on New Hardware, DaMoN 2012 (2012)
4. Chhugani, J., Nguyen, A.D., Lee, V.W., Macy, W., Hagog, M., Chen, Y.K., Baransi, A., Kumar, S., Dubey, P.: Efficient Implementation of Sorting on Multi-Core SIMD CPU Architecture. Proceedings of the VLDB Endowment 1(2) (2008)
5. Chovanec, P., Krátký, M.: Processing of Multidimensional Range Query Using SIMD Instructions. In: Abd Manaf, A., Sahibuddin, S., Ahmad, R., Mohd Daud, S., El-Qawasmeh, E. (eds.) ICIEIS 2011, Part IV. CCIS, vol. 254, pp. 223–237. Springer, Heidelberg (2011)
6. Chovanec, P., Krátký, M., Bača, R.: Optimization of Disk Accesses for Multidimensional Range Queries. In: Bringas, P.G., Hameurlain, A., Quirchmayr, G. (eds.) DEXA 2010. LNCS, vol. 6261, pp. 358–367. Springer, Heidelberg (2010)
7. Farber, R.: CUDA Application Design and Development, 1st edn. Morgan Kaufmann (2011)
8. Freeston, M.: A General Solution of the n -dimensional B-tree Problem. In: Proceedings of the ACM International Conference on Management of Data, SIGMOD 1995. ACM Press (1995)
9. Garcia, V., Debreuve, E., Barlaud, M.: Fast k Nearest Neighbor Search using GPU. In: Computer Vision and Pattern Recognition Workshops, pp. 1–6. IEEE Computer Society (2008)
10. Guttman, A.: R-Trees: A Dynamic Index Structure for Spatial Searching. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD 1984), pp. 47–57. ACM Press (June 1984)
11. Hennessy, J.L., Patterson, D.A.: Computer Architecture: A Quantitative Approach, 4th edn. Morgan Kaufmann (2006)
12. Khronos: Khronos: Opencl (2012), <http://www.khronos.org/opencl/>
13. Kirk, D.B., Mei, W., Hwu, W.: Programming Massively Parallel Processors: A Hands-on Approach. Applications of GPU Computing Series. Morgan Kaufmann (2010)
14. Krátký, M., Pokorný, J., Snášel, V.: Implementation of XPath Axes in the Multidimensional Approach to Indexing XML Data. In: Lindner, W., Fischer, F., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 219–229. Springer, Heidelberg (2004)
15. Lahdenmäki, T., Leach, M.: Relational Database Index Design and the Optimizers. John Wiley and Sons, New Jersey (2005)
16. Lightstone, S.S., Teorey, T.J., Nadeau, T.: Physical Database Design: the Database Professional's Guide. Morgan Kaufmann (2007)
17. nVIDIA: Cuda Programming Guide (2012), http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf
18. nVIDIA: nVIDIA Fermi - White Paper (2012), http://www.nvidia.com/content/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf
19. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann (2006)
20. Servetti, A., Rinotti, A., De Martin, J.: Fast Implementation of the MPEG-4 AAC Main and Low Complexity Decoder. In: Proceedings of Acoustics, Speech, and Signal Processing, ICASSP 2004 (2004)
21. Shahbahrami, A., Juurlink, B., Vassiliadis, S.: Performance Comparison of SIMD Implementations of the Discrete Wavelet Transform. In: Proceedings of Application-Specific Systems, Architecture Processors, ASAP 2005 (2005)

22. Slingerland, N., Smith, A.J.: Multimedia Extensions for General Purpose Microprocessors: A Survey. Technical report CSD-00-1124, University of California at Berkeley (2000)
23. Stonebraker, M., Abadi, D., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., Lau, E., Lin, A., Madden, S.: C-store: A Column Oriented DBMS. In: Proceedings of the International Conference on Very Large Data Bases, VLDB 2005 (2005)
24. Willhalm, T., Popovici, N., Boshmaf, Y., Plattner, H., Zeier, A., Schaffner, J.: SIMD-Scan: Ultra Fast In-Memory Table Scan Using On-Chip Vector Processing Units. Proceedings of the VLDB Endowment 2(1) (2009)
25. Zhou, J., Ross, K.A.: Implementing Database Operations Using SIMD Instructions. In: Proceedings of the ACM International Conference on Management of Data, SIGMOD 2002 (2002)

Towards Optimization of Hybrid CPU/GPU Query Plans in Database Systems

Sebastian Breß, Eike Schallehn, and Ingolf Geist

Abstract. Current database research identified the computational power of GPUs as a way to increase the performance of database systems. Since GPU algorithms are not necessarily faster than their CPU counterparts, it is important to use the GPU only if it is beneficial for query processing. In a general database context, only few research projects address hybrid query processing, i.e., using a mix of CPU- and GPU-based processing to achieve optimal performance. In this paper, we extend our CPU/GPU scheduling framework to support hybrid query processing in database systems. We point out fundamental problems and provide an algorithm to create a hybrid query plan for a query using our scheduling framework.

1 Introduction

Graphics Processing Units (GPUs) are specialized processors designed to support graphical applications, which have advanced capabilities of parallel processing and therefore nowadays have more computation power than CPUs. Using GPU to speed up generic applications is called General Purpose Computation on Graphics Processing Units (GPGPU). In particular, parallelizable applications benefit from computations on the GPU.

A new research trend focuses the acceleration of database systems by using the GPU as co-processor [1, 6, 7, 13, 15]. However, for an operation a GPU algorithm is not necessarily faster than a corresponding CPU algorithm, because the large overhead of copy operations leads to a better CPU performance for relatively small datasets [5]. A hybrid query is a query plan, which uses both, the CPU and the GPU, for the execution of a single query. To the best of our knowledge, there is no previous work on identifying problems for processing hybrid query plans in database

Sebastian Breß · Eike Schallehn · Ingolf Geist
Otto-von-Guericke University Magdeburg, Universitätsplatz 2, D-39106 Magdeburg
e-mail: [bress, eike, ingolf.geist}@iti.cs.uni-magdeburg.de](mailto:{bress, eike, ingolf.geist}@iti.cs.uni-magdeburg.de)

systems. Hence, we identify problems, which have to be solved to allow an efficient usage of the GPU as co-processor during database query processing. It is a hard task to find an optimal hybrid query plan for a given query. The need for a hybrid query optimizer was identified by Heimel [8]. We have to create all possible query plans to find the optimal plan. Furthermore, we need a cost model, which computes the cost of hybrid query plans. Then, we have to select the plan with the lowest costs. This common approach can introduce large overhead, if the optimization space is not reduced. Therefore, we use another approach. There are different approaches to estimate the execution times for single GPU algorithms [6, 11, 14]. In previous work, we presented a self-tuning decision model that was implemented as a scheduling framework and can distribute database operations response time minimal on CPUs and GPUs [3]. It is a black box approach, which computes estimated execution times for algorithms using statistical methods and observed execution times. So far, we only considered single operations. In this paper, we will present an approach how a response time minimal hybrid query plan is constructed from a logical query plan using our scheduling framework. Accordingly, the contributions provided here are first, a list of problems that occur during hybrid query processing and second a heuristic how to construct a hybrid query plan using our scheduling framework [3].

The paper is structured as follows. First, we present related work in Section 2. In Section 3, we discuss basic problems which occur during the processing and optimization of hybrid queries. We give a short overview of our decision model in Section 4 and present an approach for the construction of response time minimal hybrid query plans in Section 5.

2 Related Work

Current research investigates the use of GPUs for database operations [1, 6, 13, 15]. Walkowiak et al. discuss the usability of GPUs for databases [15] and show the applicability on the basis of an n-gram based text search engine. He et al. present the concept and implementation of relational joins on GPUs [7] and of other relational operations [6]. Pirk et al. develop an approach to accelerate indexed foreign key joins with GPUs [13]. The foreign keys are streamed over the PCIe bus while random lookups are performed on the GPU. Bakkum et al. develop a concept and implementation of the SQLite command processor on the GPU [1]. The main target of their work is the acceleration of a subset of possible SQL queries. Govindaraju et al. present an approach to accelerate selections and aggregations with the help of GPUs [4]. From this research, we conclude that a GPU is an effective co-processor for database query processing.

Ilić et al. [9] showed that large benefits for database performance can be gained if the CPU and the GPU collaborate. They developed a generic scheduling framework [10], which is a similar approach to ours, but does not consider specifics of query processing. They applied their scheduling framework to databases and tested it with

two queries of the TPC-H benchmark. However, they do not explicitly discuss hybrid query processing.

He et al. developed a research prototype, which implements relational operations on CPU and GPU, respectively [6]. They present a co-processing scheme that assigns operations of a query plan to suitable processing devices (CPU/GPU). They developed a cost model, which computes estimated execution times of single GPU algorithms in consideration of copy operations. They used a two-phase optimization model for queries: first, a relational optimizer creates an operator tree and, second, for every operator the optimizer decides if a operation is executed on GPU, CPU, or concurrently on both. He et al. proposed an exhaustive search strategy for small plans and a greedy strategy for large plans for the second phase. Since they used a calibration based method on top of an analytical cost model, their approach works currently for relational databases only, whereas our approach is more general and works with arbitrary algorithms, e.g., for XML databases. Our approach is also more general because the black-box self-adaptive mode allows the consideration of different load situations.

Heimel created the prototype *Ocelot* by implementing GPU algorithms of common relational operations in MonetDB [8]. He developed basic decision heuristics for choosing a processing unit for query execution. However, he did not consider hybrid query plans, where the CPU and the GPU are used to execute a query. Furthermore, Heimel identified two query optimizer problems. First, it is a necessity to have cost metrics, which enable the comparison of CPU and GPU algorithms. Second, the search space is bigger since placement of query plans (and hence operations) have many possibilities. Hence, he pointed out the need for a hybrid query processor and optimizer.

3 Problems during Hybrid Query Processing

The main problem of hybrid query processing is to use the GPU only if it is beneficial for the performance of a query. With new approaches of using the GPU as a co-processor for DB operations arising, the physical optimization process in database query processing has to be revised to enable an effective usage of the GPU to increase the performance of database systems. To generalize query processing from a CPU only approach to a hybrid CPU/GPU solution is a hard task. One possible approach estimates the execution times of all algorithms for an operation, choosing for each operation in a query the algorithm with the lowest estimated costs.

If a GPU algorithm is selected, then additional communication costs can arise depending on the data storage location [5]. Therefore, we need to create a set of hybrid query plan candidates and then choose the plan with lowest costs. To keep the overhead low, we have to reduce the optimization space while keeping promising candidates. Hence, we need a cost model that can compute the cost of a hybrid query plan in consideration of data storage location and possible

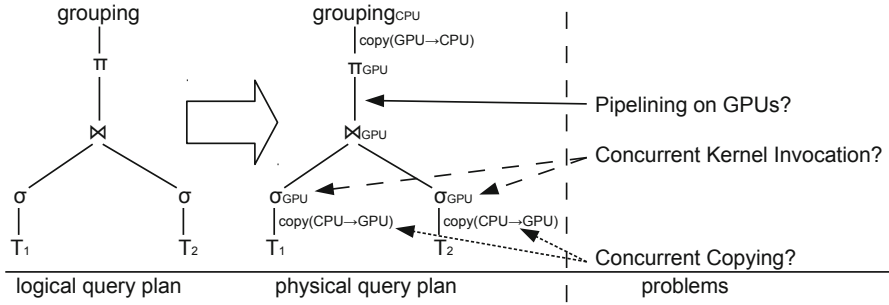


Fig. 1 Example: hybrid query plan and problems of hybrid query processing

parallel copying and data processing. We evaluate this approach in future work. Since the overhead of this approach is likely to be high, we present another solution in Section 5.

Figure 1 illustrates how a hybrid query plan is created from a logical query plan. Note the necessary copy operations, if the optimizer decides to change the processing device (CPU/GPU). We identified five problems:

1. On modern GPUs, kernels can be enqueued and processed concurrently, but inter-kernel communication is undefined [12]. Hence, a regular pipelining between two GPU algorithms is not possible. However, it is possible to integrate two operations in one kernel. Several kernels can be combined and compiled at run-time, if OpenCL is used. [8]
2. Database operations can be executed in parallel, e.g., in Figure 1, where two selections can be processed concurrently. The concurrent processing of kernels is possible for current GPUs [12], but it is hard to predict the influence on execution times.
3. Concurrent copy operations in the same direction are not possible [12], but occur in Figure 1 before the selections on the GPU are performed. Hence, the copy operations need to be serialized, and the following selections have to be serialized as well. A possible approach would be to combine two data streams in one copy operation and reorganize the data in the GPU RAM. This can be faster, because the PCIe Bus is better utilized.
4. Since the number of concurrent kernel executions (16 by current NVIDIA GPUs [12]) and the PCIe Bus bandwidth are limited, not every query benefits from the GPU. Thus, a heuristic is needed, which chooses "critical queries" that first benefit from the GPU usage and second have a certain degree of "importance", because some queries need higher performance than others.
5. A further problem is to estimate how the optimization of one query influences the performance of another hybrid query. Since this is hard to predict, we do not consider it here and address it in future work.

4 Decision Model

4.1 Overview of the Model

To decide about the optimal processing unit (CPU/GPU) we collect observations of past algorithm executions and use statistical methods to interpolate future execution times [2], [3]. Let O be a database operation and let $AP_O = \{A_1, \dots, A_m\}$ be an algorithm pool for operation O . The algorithms in AP_O are executable on CPU or GPU. We assume that every algorithm can be faster than the other algorithms in AP_O depending on the dataset the operation is applied on. Let $T_{est}(A, D)$ be an estimated and $T_{real}(A, D)$ a measured execution time of algorithm A for a dataset D . Then MPL_A is a measurement pair list, containing all current measurement pairs $(D, T_{real}(A, D))$ of algorithm A . Based on the collected measurements, an estimation component provides estimations for each algorithm for a requested operation. The component uses an approximation function $F_A(D)$ derived from MPL_A to compute the estimations $T_{est}(A, D)$. Accordingly, a decision component chooses the algorithm that fits best with the specified optimization criteria. Figure 2 summarizes the model structure.

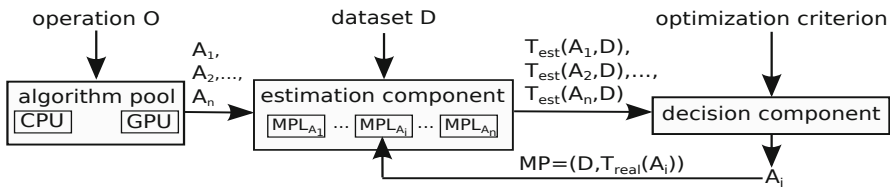


Fig. 2 Overview of the decision model

4.2 Decision Component

In the work presented in this paper, only the optimization of the response time is discussed, i.e. selecting the CPU- or GPU-based algorithm with the minimal estimated execution time for a dataset D . Using our approach it is possible to automatically fine-tune algorithm and, accordingly, processing unit selection on a specific hardware configuration at run-time. The decision model is generic, i.e., no prior knowledge about hardware parameters or details about used algorithms is required at development time.

5 Construction of Hybrid Query Plans from Logical Query Plans

We present an approach how a good but not optimal hybrid query plan is generated using our decision model. We assume for simplicity that a (logical) query plan is a sequence of operations $Q_{log} = O_1 O_2 \dots O_n$. We construct a hybrid query plan

by choosing for each operation O_i in Q_{log} the response time minimal algorithm, which leads to a hybrid query plan Q_{hybrid} . Depending whether an algorithm uses the CPU or GPU, the operation is executed on the corresponding processing unit. Let $CA(D, O)$ be a function, which chooses the fastest algorithm A for a given data set D and an operation O . It uses the function T_{est} to compute estimated execution times for algorithms. T_{est} considers the time needed to copy data to and from the GPU RAM in the case a GPU algorithm is chosen. Hence, $CA(D, O)$ chooses a GPU algorithm only, if the execution time of a CPU algorithm is greater than the execution time of a GPU algorithm plus the time needed for the data transfers. Let $CAS(A)$ be a function, that returns an algorithm sequence needed to execute algorithm A . In case of a CPU algorithm, $CAS(A)$ returns A . In the case of a GPU algorithm, $CAS(A)$ returns a sequence of three algorithms. The first is $A_{cpy}(D, HD)$, which copies the input data from the CPU RAM to the GPU RAM (host to device). The second is $(A_{i,GPU}(D))$, which processes the data set D on the GPU. The third is $A_{cpy}(O(D), DH)$, which copies the result set back to the CPU RAM (device to host). In case of a CPU algorithm, operation O_i is substituted by $A_{i,CPU}(D)$.

$$T_{est}(D, A) = \begin{cases} T_{est}(D, A) & \text{if } A = A_{CPU} \\ T_{est}(A_{cpy}(D, HD)A(D)A_{cpy}(O(D), DH)) & \text{otherwise} \end{cases} \quad (1)$$

$$CA(D, O) = A \text{ with } T_{est}(D, A) = \min\{T_{est}(D, A) | A \in AP_O\} \quad (2)$$

$$CAS(A) = \begin{cases} A(D) & \text{if } A = A_{CPU} \\ A_{cpy}(D, HD)A(D)A_{cpy}(O(D), DH) & \text{otherwise} \end{cases} \quad (3)$$

We formalize our approach in algorithm [□](#). In lines 1–6 we construct the optimal query plan using the functions $CA(D, O)$ and $CAS(A)$ of our decision model by choosing the fastest expected algorithm for each operation in a query. The algorithm leads to two succeeding copy operations in different directions, when two succeeding operations shall be executed on the GPU. This needless copy operations are removed by the algorithm in lines 7–11.

Example: For the following example, we omit the datasets in the algorithm notation. We consider selections (S), projections (P), joins (J), and groupings (G). The query plan from Figure [□](#) is written like this: $O_S O_S O_J O_P O_G$. After the first loop of the algorithm was processed, the result is:

$$A_{S,CPU} A_{S,CPU} A_{cpy}(HD) A_{J,GPU} A_{cpy}(DH) A_{cpy}(HD) A_{P,GPU} A_{cpy}(DH) A_{G,CPU}$$

After the second loop of the algorithm was processed the result is:

$$A_{S,CPU} A_{S,CPU} A_{cpy}(HD) A_{J,GPU} A_{P,GPU} A_{cpy}(DH) A_{G,CPU}$$

Since the decision model decided to use a GPU algorithm in two cases, we can assume that the response time of the hybrid plan is smaller than the time of the pure CPU plan. If the optimizer expects that no GPU algorithm will be faster with respect

Algorithm 1. Construction of hybrid query plan Q_{hybrid} from logical query plan Q_{log}

Input: $Q_{log} = (O_1, D^1); \dots; (O_n, D^n)$

Output: $Q_{hybrid} = A_1 \dots A_m$

1. $Q_{hybrid} = \emptyset$
 2. **for** O_i in Q_{log} **do**
 3. $A = CA(D^i, O)$
 4. $AS = CAS(A)$
 5. append AS to Q_{hybrid}
 6. **end for**
 7. **for** A_i in Q_{hybrid} **do**
 8. **if** ($A_i = A_{cpy}(D, DH)$ **and** $A_{i+1} = A_{cpy}(D, HD)$) **then**
 9. delete $A_i A_{i+1}$ from Q_{hybrid}
 10. **end if**
 11. **end for**
-

to the examined query, than the resulting query plan and a pure CPU query plan are equal, i.e., the generated plan uses only the CPU.

Reflection on the Algorithm: Our proposed algorithm does not generate an optimal hybrid query plan in all cases, because the algorithm uses a greedy strategy. We consider for the cost computation no concurrent copying and processing and hence, sum up the times of all algorithms in a plan to compute the execution time of a query plan. In this example, we will use the execution times shown in Table 1. Consider the query plan $O_S O_S O_J O_P O_G$ and assume the algorithm processes O_J . Then $T_{est}(A_{cpy}(HD)A_{J,GPU}A_{cpy})$ is greater than $T_{est}(A_{J,CPU})$ ($3 + 2 + 3 = 8 > 5$) and the algorithm decides for the CPU algorithm for the Join. However, if the algorithm had considered O_J and the successor O_P , then it would have seen that $T_{est}(A_{cpy}(HD)A_{J,GPU}A_{P,GPU}A_{cpy}(DH))$ is less than $T_{est}(A_{J,CPU}A_{P,CPU})$ ($3 + 2 + 1 + 3 = 9 < 5 + 5$), so the usage of the GPU algorithms for the join and the projection would result into a cheaper query plan. Since the algorithm only chooses locally optimal solutions and does not look forward in the query plan, it cannot consider the possibility that the selection of a slower algorithm could lead to a faster query plan, because it cannot foresee the copy optimizations. However, the algorithm is able to create a promising candidate, where, e.g., an evolutionary algorithm can be applied to find better plans through mutations and crossover. We believe that our algorithm can be a basis for further optimizations.

Table 1 Example execution times of algorithms for the given example datasets

processing unit	O_S	O_J	O_P	O_G	$O_{cpy}(HD)$	$O_{cpy}(DH)$
CPU Time	1	5	5	2	3	3
GPU Time	3	2	1	7	-	-

6 Future Work

To address the problem of parallel processing of different queries, we will present a heuristic that will decide which database queries can benefit most from using the GPU, because not all queries can benefit from GPU co-processing. Therefore, we will compute the cost of a pure CPU query and a hybrid query and compute the gain, which a GPU co-processing would have for the query. Therefore, we need cost metrics, which compute the cost of a hybrid query plan.

An alternative approach to deal with parallelism within and between queries would be to allow both by default, and let the GPU schedule parallel requests on its own. As pointed out in Section 3, execution times will be harder to estimate and the benefit for single queries will decline. Nevertheless our self-learning cost-estimation will adjust to this and can find a balance, because estimated execution times will increase due to concurrency situations, and only queries benefiting most from a GPU-based execution will be executed as hybrid queries based on our described decision model. This approach has to be carefully evaluated.

Since our algorithm does not generate an optimal plan in all cases, other solutions have to be considered. Another approach to find the cheapest query plan would be to generate a candidate set of hybrid query plans, and apply a cost metric to each of them and then choose the cheapest plan for execution. The possible benefit and overhead of this according approaches will be examined in future work. Furthermore, we will implement our framework in Ocelot, a research prototype for using GPU algorithms in databases [8] and validate our heuristic.

7 Conclusion

In this paper, we pointed out common problems that occur during the optimization of hybrid query processing and need to be addressed to allow an effective co-processing by the GPU during database query processing. Furthermore, we provided a simple algorithm for constructing a good hybrid query plan for a given logical query using our scheduling framework.

Acknowledgements. The work in this paper has been funded in part by the German Federal Ministry of Education and Science (BMBF) through the Research Program under Contract No. FKZ: 13N10817. We thank Mario Pukall and Maik Mory for helpful feedback and discussions.

References

1. Bakkum, P., Skadron, K.: Accelerating SQL database operations on a GPU with CUDA. In: Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units, GPGPU 2010, pp. 94–103. ACM, New York (2010)

2. Breß, S., Beier, F., Rauhe, H., Schallehn, E., Sattler, K.U., Saake, G.: Automatic Selection of Processing Units for Coprocessing in Databases. In: 16th East-European Conference on Advances in Databases and Information Systems, ADBIS. Springer (2012)
3. Breß, S., Mohammad, S., Schallehn, E.: Self-Tuning Distribution of DB-Operations on Hybrid CPU/GPU Platforms. In: Grundlagen von Datenbanken, pp. 89–94. CEUR-WS (2012)
4. Govindaraju, N.K., Lloyd, B., Wang, W., Lin, M., Manocha, D.: Fast Computation of Database Operations using Graphics processors. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD 2004, pp. 215–226. ACM, New York (2004)
5. Gregg, C., Hazelwood, K.: Where is the data? Why You Cannot Debate CPU vs. GPU Performance without the Answer. In: Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, ISPASS 2011, pp. 134–144. IEEE Computer Society, Washington, DC (2011)
6. He, B., Lu, M., Yang, K., Fang, R., Govindaraju, N.K., Luo, Q., Sander, P.V.: Relational Query Coprocessing on Graphics Processors. *ACM Trans. Database Syst.* 34, 21:1–21:39 (2009)
7. He, B., Yang, K., Fang, R., Lu, M., Govindaraju, N., Luo, Q., Sander, P.: Relational Joins on Graphics Processors. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, pp. 511–524. ACM, New York (2008)
8. Heimel, M.: Investigating Query Optimization for a GPU-accelerated Database. Master's thesis, Technische Universität Berlin, Electrical Engineering and Computer Science, Department of Software Engineering and Theoretical Computer Science (2011)
9. Ilić, A., Pratas, F., Trancoso, P., Sousa, L.: High-Performance Computing on Heterogeneous Systems: Database Queries on CPU and GPU. In: High Performance Scientific Computing with Special Emphasis on Current Capabilities and Future Perspectives, pp. 202–222. IOS Press (2011)
10. Ilić, A., Sousa, L.: Chps: An environment for collaborative execution on heterogeneous desktop systems. *International Journal of Networking and Computing* 1(1) (2011)
11. Kothapalli, K., Mukherjee, R., Rehman, M.S., Patidar, S., Narayanan, P.J., Srinathan, K.: A performance prediction model for the CUDA GPGPU platform. In: 2009 International Conference on High Performance Computing, HiPC, pp. 463–472 (June 2009)
12. NVIDIA: NVIDIA CUDA C Programming Guide, Version 4.0, pp. 30–34 (2012), http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf (accessed May 1, 2012)
13. Pirk, H., Manegold, S., Kersten, M.: Accelerating Foreign-Key Joins using Asymmetric Memory Channels. In: VLDB - Workshop on Accelerating Data Management Systems Using Modern Processor and Storage Architectures (ADMS): Proceedings of International Conference on Very Large Data Bases 2011 (VLDB), pp. 585–597 (2011)
14. Schaa, D., Kaeli, D.: Exploring the multiple-GPU design space. In: Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing, IPDPS 2009, pp. 1–12. IEEE Computer Society, Washington, DC (2009)
15. Walkowiak, S., Wawruch, K., Nowotka, M., Ligowski, L., Rudnicki, W.: Exploring Utilisation of GPU for Database Applications. *Procedia Computer Science* 1(1), 505–513 (2010)

Thrust and CUDA in Data Intensive Algorithms

Krzysztof Kaczmarek and Paweł Rzażewski*

Abstract. Huge memory bandwidth and instruction throughput make GPU processors very attractive for many algorithms which can only utilize Single Instruction Multiple Data (SIMD) architecture. Databases and their data intensive operations may also benefit from parallel GPU threads and thread streams. Many libraries offer simple interfaces for GPU, which make memory and threads management as easy as possible. Trade-off in programmers' time, code structure and algorithm efficiency is critical for business applications. In this paper we evaluate the usage of Thrust library and its ability to manage millions of threads when compared to pure CUDA C program.

1 Introduction

General Purpose Graphic Processor Unit (GPGPU) programming, allowing significant acceleration in various algorithms, focused great attention of wide research community, including databases. Lots of applications already demonstrated even two orders of magnitude speed-up, when compared to classical CPU-based sequential algorithms. However, in case of databases, we face two problems, which are hard to solve in GPU processing. First, GPU working in single-instruction-multiple-data processing model requires all data to be of uniform size and structure. Any deviations or uncertainties may significantly decrease the performance. Secondly, each thread should follow the same execution path. This makes GPU useful at low level and only for simply structured data. We already evaluated application of GPU processor in MOLAP databases with excellent results [4].

Krzysztof Kaczmarek · Paweł Rzażewski
Warsaw University of Technology, Faculty of Mathematics
and Information Science, ul. Koszykowa 75, 00-662 Warsaw, Poland
e-mail: [k.kaczmarek,p.rzazewski}@mini.pw.edu.pl](mailto:{k.kaczmarek,p.rzazewski}@mini.pw.edu.pl)

* Corresponding author.

Typical application of GPU threads in data intensive applications tends to bind a single thread with a single data record or a set of records. This approach may lead to enormous number of threads executed and high utilization of scheduling mechanism. Additionally, the data intensive applications often suffer from a limited capacity of memory bandwidth, slowing down the communication between the processor and memory. Hiding this memory latency is possible only when a processor may switch between memory reading or writing and arithmetic operations. Another aspect of efficient GPU programming is a proper memory access organization in order to allow coalesced memory access and avoid shared memory bank conflicts. These tasks are not easy if data records change size dynamically.

Fortunately there are more and more libraries addressing the highlighted problems. A programmer working on higher level may easily focus on algorithm, not wasting time for tricky details which can be also cumbersome. Another great advantage is an independence of a certain hardware and write-once-run-everywhere solution. Motivation of our work was to evaluate a high level library application for a data intensive algorithm and check its efficiency on different platforms. We choose Thrust [3] as a new library and an emerging standard for high level GPU programming, similar to C++ STL library. The task we selected is permutation generating, which is not a pure database problem but it covers aspects which are important for any database system:

1. many memory read and write operations
2. huge amount of data
3. several possible implementations varying in fitting to GPU
4. no constant number of threads or iterations

The rest of the paper is organized as follows. We describe the problem to be solved in section 2 and several implementations for Thrust and CUDA in section 3. Experiments are described in section 4 while the discussion of the results is covered in section 4.1 and 5.

2 Task Description

Generating permutations of a given set is a hard problem, with complexity exceeding any computer capabilities even for small sets. Therefore it is good to push GPU and both low-level and high-level API to the limits.

There are many approaches to generating all permutations of a given set (see for example [5, 9, 11]), differing in order of permutations or memory requirements. To simulate memory intensive operations, we generate and store all permutations of a set $[n]$ in memory.

For this experiment, we developed two algorithms which can be executed in parallel threads – we call them *Bottom-up* and *Top-down*. In the following

¹ For a natural number n , by $[n]$ we denote the set $\{0, 1, \dots, n - 1\}$.

subsections we give pseudo-codes and short descriptions of both algorithms. Note that in pseudo-codes all vectors are zero-indexed.

2.1 Bottom-Up

The algorithm computes the table *perm* of all permutations of $[n]$, using the dynamic programming approach. The algorithm starts with storing the permutation $(0, 1, \dots, n-1)$ in *perm*[0]. Then for $i = 1, 2, \dots, n-1$ the algorithm constructs and stores all permutations p , such that $i = \max\{s: p_s \neq s\}$ (note that $i \geq 1$ since it is impossible to have only one s such that $p_s \neq s$).

Suppose we have generated all permutations for i up to some $i_0 - 1$ (there are exactly $(i_0 - 1)!$ of them). The permutations for i_0 are generated as follows. Take the k -th permutation (for $k \in [(i_0 - 1)!]$) and swap the i_0 -th element (equal to i_0) with every possible element from positions $0, \dots, i_0 - 1$. Each of the permutations generated in this way is stored in the table *perm*.

Here is an example to clarify the idea. Let $n = 3$. The algorithm starts with the permutation $(0, 1, 2)$, which is stored in *perm*[0].

To generate the permutations for $i = 1$ we take the permutation $p = (0, 1, 2)$ and swap the element $p_1 = 1$ with the element p_0 , obtaining the permutation $p' = (1, 0, 2)$ and storing it in *perm*[1].

This is the only permutation of $\{0, 1, 2\}$ for $i = 1$, so i is increased to 2. Then we take k -th permutation (for $k = 0, 1 = (i - 1)! - 1$) and store the permutation obtained by swapping the element 2 with every element in positions 0 and 1. This way we obtain the remaining permutations.

Algorithm [□](#) presents the pseudo-code of the algorithm *Bottom-up*.

One can easily observe that after the execution of the algorithm, the table *perm* contains all permutations of $[n]$.

For each $i = 1, \dots, n - 1$ the algorithm has to perform $i!$ *read* operations, $i! \cdot i$ *swap* operations and $i! \cdot i$ *write* operations. All those operations can be performed in parallel. Therefore the complexity of the algorithm *Bottom-up* is $T_{BU}(n) = \sum_{i=1}^{n-1} \left(\sum_{k=0}^{i!-1} (1 + \sum_{j=0}^{i-1} 2) \right) / p = \sum_{i=1}^{n-1} \frac{(1+2i)i!}{p} = \sum_{i=1}^{n-2} \frac{(1+2i)i!}{p} + \frac{(2n-1)(n-1)!}{p} = \Theta\left(\frac{n!}{p}\right)$, where p is the number of processors available.

0	perm (0, 1, 2)	0	perm (0, 1, 2)	0	perm (0, 1, 2)
		1	(1, 0, 2)	1	(1, 0, 2)
				2	(2, 1, 0)
				3	(0, 2, 1)
				4	(2, 0, 1)
				5	(1, 2, 0)

Fig. 1. Table *perm* created by the main loop (lines [□-□](#)). From the left: before the first iteration, before the second iteration, after the second iteration.

Algorithm 1. *Bottom-up*(int n)

```

1 perm[0] ← (0, 1, 2, 3, 4, ..., n - 1)
2 for i ← 1 to n - 1 do
3   for k ← 0 to i! - 1 do in parallel
4     p ← perm[k]
5     for j ← 0 to i - 1 do in parallel
6       pj ↔ pi
7     perm[i! + k · i + j] ← p

```

2.2 Top Down

The algorithm *Top-down* is very similar to the previous one. It generates k -th permutation of the set $[n]$ (for $k \in \{0, \dots, n! - 1\}$). Again, we shall demonstrate the idea on an example. Let us generate the 19th permutation of the set $\{0, 1, 2, 3\}$ (so $k = 19$ and $n = 4$).

We start with $p = (0, 1, 2, 3)$. Let i be a maximum integer such that $i! \leq k$, j be equal to $k \bmod i$ (in this case $i = 3$ and $j = 1$). We swap the elements $p_i = 3$ and $p_j = 1$, obtaining $p = (0, 3, 2, 1)$ and update k to $\frac{k-i!-j}{i} = 4$. We repeat the process until $k = 0$.

```

Step 1: k=19 p=(0, 1, 2, 3) i=3 j=1
Step 2: k=4  p=(0, 3, 2, 1) i=2 j=0
Step 3: k=1  p=(2, 3, 0, 1) i=1 j=0
Step 4: k=0  p=(3, 2, 0, 1)

```

Fig. 2. Steps of computation of 19th permutation of $\{0, 1, 2, 3\}$ using the algorithm *Top-down*

The order of permutations is different than the one for the algorithm *Bottom-up*. Algorithm 2 presents the pseudo-code of the algorithm *Top-down*.

Observe that always $i \leq n - 1$ and in each step the value of i is decreased. Therefore the complexity of the algorithm *Top-down* is $T_{TD}(n) = \Theta(n)$.

Algorithm 2. *Top-down*(int k, int n)

```

1 p ← (0, 1, 2, ..., n - 1)
2 while k > 0 do
3   i ← max{h: h! ≤ k}
4   j ← k mod i
5   k ← (k-i!-j)/i
6   pj ↔ pi
7 return p

```

3 Coding Experiences

Both algorithms were implemented in Thrust and CUDA C. Since many database applications work on records which cannot be represented as a structure of arrays, we decided keep the array *perm* as a typical array of structures. In our case, these structures are simple byte arrays. We observe that this data model is not optimal since length of array changes with n , which may lead to non-coalesced reads or unpredictable bank conflicts. However, we keep it as a good exemplar of typical database application.

3.1 Thrust Library

Thrust is a high level API which mimics STL C++ library. It works on `device_vector` and `host_vector` objects from which the first one is stored on the GPU device side and the second one resides in the CPU RAM. Device side programming is much simplified by many useful operators and utility functions. Also the basic parallel algorithms like reduction, scan, sorting and others are implemented. The programmer is free in switching to classical CUDA C programming by conversions between normal C pointers and Thrust objects. However, the most useful tools in Thrust are iterators, combined with custom device functions. Many tasks may be simply expressed by a combination of several iterators, among which the most important are permutation, counting, constant and zip. The last one improves the performance of GPU programs by assuring a support for structure-of-arrays (SOA) instead of array-of-structures (AOS) which is more popular and much simpler.

3.1.1 Bottom-up Algorithm

The *Bottom-up* algorithm is based on a permutation iterator which gets all the permutations from the previous step i and produces $i + 1$ new permutations for each input permutation. One thread reads one permutation and writes one new permutation. In step i , we need $(i + 1)! - i! = i! \cdot i$ threads. Each new permutation, created by one thread, is defined by the number of thread (t), the number of input permutation from the previous step (p), the number of current step (i) and the number of exchange position (j).

In Thrust we have no control on threads allocation and any other low level CUDA feature. Also we can only define a device function which processes explicitly passed parameters. It cannot access a state of other threads. This property simplifies parallelism and lets to run many threads without any additional synchronization, but also makes the implementation of algorithms more difficult or sometimes even impossible.

In our *Bottom-up* procedure we need to pass the necessary state of the algorithm into each thread. We may achieve this by composing four iterators:

1. constant iterator for i
2. counting iterator for $t = 0, 1, \dots, (i + 1)! - i! - 1$
3. transformation iterator for $p = t \text{ div } i!$
4. transformation iterator for $j = t \text{ mod } i!$

The idea of iterators and how they are used is presented in fig.3. The collection of already generated permutations is expanded by a set of new permutations. All four iterators are known before the threads launch, so each thread may work independently of any other thread.

	perm	t p i j		perm	t p i j
0	(0, 1, 2, 3)	3	(0, 2, 1, 3)	1 0 2 1
1	(1, 0, 2, 3)	4	(2, 0, 1, 3)	2 1 2 0
2	(2, 1, 0, 3)	0 0 2 0	5	(1, 2, 0, 3)	3 1 2 1

Fig. 3. Visualization of Thrust iterators for $i = 2$ and resulting generated permutations. Permutations created in the previous step are marked by dots.

3.1.2 Top-down Algorithm

The *Top-down* algorithm is much simpler for implementation than the *Bottom-up* one. The procedure generating each permutation does not depend on any other permutation. The only input is the permutation number.

If we assign a single thread to generate a single permutation, we get an embarrassingly parallel problem. However, it is not well suited for GPU threads. A general paradigm of GPU programming [6, 8] states that all threads within a block should perform exactly the same execution path and the same number of loop iterations. We can notice that in the *Top-down* algorithm (alg. 2) the loop iterates different number of times for different permutations (and therefore for different threads). This significantly slows down the execution and by any means cannot be solved.

3.2 CUDA C

CUDA C [7] is the basic programming interface for GPGPU processors developed by NVIDIA. It allows for a much more fine-grained control of the threads execution as well as memory allocation and deallocation. A programmer using CUDA must be familiar not only with many low level rules and requirements, but also with different hardware architectures to get optimized and scalable applications.

The *Bottom-up* procedure for CUDA C can be implemented in two different ways. The first one may be a straightforward implementation of the algorithm [1](#). Actually, if we consider the loop in line 3 to be allocated to parallel threads then the inner loop in line 5 cannot be further split into new threads. In CUDA, a kernel procedure may only call single threaded device functions. Therefore each thread in step i has to read one permutation and write sequentially i permutations. In each step i we run $i!$ threads. The second option for CUDA C is to follow exactly the same philosophy as in case of Thrust, replacing explicit iterations by indexes passed to the kernel as in case of iterators. The values of p , i and j are calculated from a thread number. Each thread reads one permutation and writes one permutation. In step i we need i times more threads than in the previous case which gives the total number of $i! \cdot i$ threads.

The *Top-down* procedure in CUDA C is implemented exactly in the same way as in Thrust. In this case, kernel function is executed on $i!$ threads and performs exactly the procedure from the algorithm [2](#). The same inefficient behaviour of this procedure applies.

In both approaches, we had to find an optimal number of threads in blocks. Due to different hardware capabilities and different requirements, it was changing for different devices. This time consuming optimization phase was important for CUDA C programs and finally reduced execution times to values comparable or even better than in Thrust.

3.3 Summary

For the experiments, seven implementations were prepared: *Bottom-up* algorithm using: Thrust; CUDA C (with iterations inside threads marked as '1'); CUDA C (without iterations marked as '2'); C++ STL (on CPU processor). *Top-down* using: Thrust; CUDA C; C++ STL (on CPU processor).

This setting lets us to compare a behaviour and scalability of Thrust and CUDA not only for various data load, but also for different hardware. CPU procedure is added just for testing and information purposes.

4 Runtime Experiments Results

All GPU implementations were run on two distinct devices, which gives altogether 12 experiments. In each experiment we generated permutations for $n = 6, \dots, 11$. Figure [4](#) presents the runtime experiments results. In order to analyse the efficiency of Thrust, CUDA C and CPU, we run the described procedures with the following configurations:

- CPU: Intel’s 3Ghz CPU Core2 Duo E8400, 6MB cache, 4GB of RAM
- GPU 1: NVIDIA GTX 280 1.242 GHz, 240 cores 1GB DDR3 (CC² 1.3)
- GPU 2: NVIDIA Tesla 2050 1.15GHz, 448 cores, 3GB DDR5 (CC 2.0)

Throughout all the experiments we used CUDA 4.0 and Thrust 1.4 running on Ubuntu 10.04 LTS operating system.

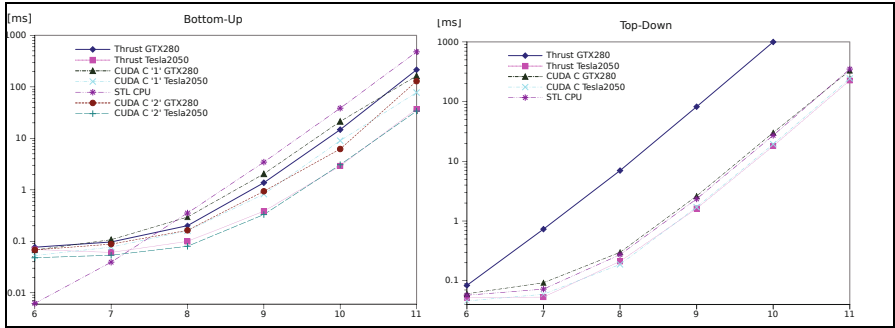


Fig. 4. Time of generating all permutations of $[n]$ for $n = 6, \dots, 11$

4.1 Discussion

The experimental results are very interesting. First of all, we can easily notice that Tesla device is really much faster than GTX device and the factor varies from 5.7 in Thrust for well suited *Bottom-up* algorithm, through 2.1 and 3.8 in CUDA C, up to 1.35 in badly conditioned *Top-down* approach in CUDA C. Very strange situation can be observed in *Top-down* algorithm running on GTX device. It is about 58 times slower than the same code running on Tesla. We repeated the experiments several times and the results were always the same, proving that this case is really ill-conditioned for older GPUs. Details of this behaviour are described later during profiler output discussion.

The second observation is that generally Thrust handles Tesla (Fermi type) devices much better than the older ones. Table I presents times for the *Bottom-up* algorithm in CUDA C and Thrust for both devices. We can see that while Thrust is significantly slower on the older device, it speeds up for Tesla and its results are comparable to pure CUDA, which is an excellent result considering that it is really a high-level library.

Analysis of processor occupancy and thread block configuration shows that for large vectors Thrust tries to call a device function multiple times in single thread, choosing an optimal configuration of threads for a given number of streaming processors. Despite better occupancy (0.65 versus 0.5 in case of CUDA C), it achieves worse results, because of less shared memory allocated

² CUDA Computation Capability.

Table 1. Comparison of time results in milliseconds of the same implementation of *Bottom-up* algorithm running on different hardware

n	Thrust GTX	Thrust Tesla	CUDA C GTX	CUDA C Tesla
9	1.3725	0.3824	0.9365	0.3303
10	14.7368	2.9119	6.2003	3.0717
11	214.5940	37.0736	129.0200	33.8290

per block. This strategy is even worse in case of *Top-down* algorithm. Since each call of the permutation computing function may iterate different number of times, one thread may follow much different branch of execution than a neighbouring thread. This must degrade performance and is observable in the figure [4](#) on the right.

5 Conclusions and Related Works

From the data intensive algorithms point of view, the achieved results are very promising. A procedure running on Tesla device and implemented in Thrust, even if not following all the recommended practices for GPU programming, is still more than 100 times faster than on CPU. Thrust proved to be efficient and very close to low level programming in CUDA C for a Fermi device but a little less efficient for 1st generation CUDA devices. Thrust automatically scored the best speed-up when moving from the old to the new device architecture. Lower level CUDA C programs must be manually tuned in order to utilize all available power in a similar situation.

There are not many similar comparisons of Thrust and CUDA in the case of data intensive applications. The authors of Thrust [\[1\]](#) explain that it performs low level fine-tuning of kernel launch parameters in order to get the highest possible occupancy. This kind of optimizations are often well-known but rarely implemented due to variety of target hardware platforms they should cover. Therefore for example array filling procedure is usually done in a naive way. The authors report up to 34 times faster run with Thrust working on small data types and older devices.

Authors of the Chestnut language [\[10\]](#), while evaluating their solution, compare Thrust to highly optimized pure CUDA code. In case of sort operation CUDA is around 3 times faster, map times are the same while in case of reduction CUDA is 58 times faster. These experiments were performed on a rather old device with only 48 CUDA cores. We expect much smaller differences for Fermi architecture and new Thrust version. Although authors suggest that Thrust is too complicated for an average user, we do not agree. Thrust is not more complicated than STL and Boost [\[2\]](#), which already gained a large community of professional users in industry and science.

Our experiment showed that Thrust may be effectively used as a high level programming library for the data intensive applications. It was able to perform not worse than a pure CUDA code. In the same time, the advantages of Thrust utilization in database systems are hard to overestimate. Productivity, abstractness and modularity allow very same Thrust functors to be reused in many new tasks. GPU devices with CC lower than 2.0 may benefit from manually optimized pure CUDA code. This strategy is proposed by Thrust authors who claim that there are no universal tool for everything.

References

1. Bell, N., Hoberock, J.: Thrust: A Productivity-Oriented Library for CUDA. In: GPU Computing Gems Jade Edition. Morgan Kaufmann (2011)
2. Dawes, B., Abrahams, D.: Boost C++ Libraries (2012), www.boost.org
3. Hoberock, J., Bell, N.: Thrust CUDA Library v.1.4.0 (2011)
4. Kaczmariski, K., Rudny, T.: MOLAP Cube Based on Parallel Scan Algorithm. In: Eder, J., Bielikova, M., Tjoa, A.M. (eds.) ADBIS 2011. LNCS, vol. 6909, pp. 125–138. Springer, Heidelberg (2011)
5. Knuth, D.: The art of computer programming, generating all tuples and permutations, fascicle 2, vol. 4 (2005)
6. NVIDIA Corporation: CUDA C best practices guide (2011)
7. NVIDIA Corporation: CUDA C Toolkit and SDK v.4.0 (2011)
8. NVIDIA Corporation: NVIDIA CUDA C programming guide version 4.0 (2011)
9. Sedgewick, R.: Permutation generation methods. ACM Comp. Surv., 137–164 (1977)
10. Stromme, A., Carlson, R.: Chestnut: Simplifying General Purpose Graphics Processing. Technical Report (2010), www.wsrn.sccs.swarthmore.edu/users/11/rcarlso1/docs/RyanCarlson_parallel.pdf
11. Tsay, J.C., Lee, W.P.: An optimal parallel algorithm for generating permutations in minimal change order. Parallel Comput. 20(3), 353–361 (1994), [http://dx.doi.org/10.1016/S0167-8191\(06\)80018-9](http://dx.doi.org/10.1016/S0167-8191(06)80018-9), doi:10.1016/S0167-8191(06)80018-9

Part II
Mining Complex and Stream Data

A Detection of the Most Influential Documents

Dariusz Ceglarek and Konstanty Haniewicz

Abstract. This work is a result of the ongoing research on semantic compression and robust algorithms applicable in plagiarism detection. This article includes a brief description of Sentence Hashing Algorithm for Plagiarism Detection SHAPD along with a comparison with the other available alternatives using frame structures for subsequence detection. What is more, the core of this publication is devoted to the application of SHAPD to a task of discovery of the most influential documents in a corpus. The experiments were carried out on multiple datasets diversified in terms of structure and content. The observations gathered during the experiments were summarised and are given in the article. The experiment allowed the authors to verify their initial hypothesis that it is possible to single out the most important documents in a corpus capturing the relations of citation among them.

1 Introduction

Thanks to the already obtained results from the previous experiments it was possible to use Sentence Hashing Algorithm for Plagiarism Detection SHAPD [6] in a new application. Its main premise is to test whether it is possible to identify the most influential documents in a given corpus. The relations enabling for singling out of the most influential documents account for:

- number of common subsequences with other documents,
- self-citation without providing a reference of given document,
- commonly used phrases in given document,
- number of documents citing given document,

Dariusz Ceglarek
Poznan School of Banking, Poznan, Poland
e-mail: dariusz.ceglarek@wsb.poznan.pl

Konstanty Haniewicz
Poznan University of Economics, Poznan, Poland
e-mail: konstanty.haniewicz@ue.poznan.pl

- number of documents cited by given document,
- number of subsequences being an instance of plagiarism.

In spirit of the above-given relations, a document that is to be classified as an influential one, is one that shares one or both major traits:

- is heavily cited in the processed corpus,
- cites a considerable number of heavily cited documents from the processed corpus.

This working definition of the influential documents originates in observations of a large number of tools readily available to users on the WWW. Most of them, display a number of characteristics such as a citation count and a number of indices. These indices, an example of which is a h-index [11], provide a measure that informs a user of a perceived value of author/work to the specific community. There are no clear qualitative limits on the enumerated relations, as the proposed method operates on the user provided corpus. Therefore, the influential documents are those that rank the highest among documents in processed corpus.

The postulated functionality of delivering a set of influential documents in a given corpus, might be achieved by already existing solutions not dealing with the content of a document but only with its annotated abstracts and bibliographic data. Bibliographic solutions provide a wide array of indices, yet they usually omit the second enumerated trait. This trait is introduced to provide data on documents that try to provide an overview of a given domain represented by the processed corpus.

The advantage of the described method over bibliographic solutions lays in its independence from various formats used in documents and additional input obtained from the external users. The proposed method processes documents in a fully automated manner and uses SHAPD to analyse the actual textual data to provide the conclusions used to classify a given document as an influential one.

As will be discussed in the related works section, the common sequences can be given by competing solutions yet there are important issues such as the length of a text frame (its variance and the situations where it becomes long) and resilience to local discontinuities of a matched sequence that are handled much better by SHAPD. Of importance is also the performance of the used algorithm that is considerable better than the w-shingling [15, 4] based solutions and dramatically better when it comes to solutions such as Smith-Waterman [13] algorithm.

Of utmost importance is the fact that SHAPD is an algorithm that **was not designed** to solve the problem of document overlapping neither was it designed to find document duplicates. It was designed to robustly compute the longest common subsequences in text documents. The robustness is based on the observation that a sentence is the most important elemental particle of a text. As sentences vary in length, shingle oriented algorithms perform worse than SHAPD. It is an effect of a choice of suboptimal structure for the representation of a problem.

The experiments used corpora of scientific articles originating from a number of distinct domains. Their initial structure varied due to their origin and distinct characteristics. SHAPD proved of great use in overcoming those dissimilarities and computing the values associated with the previously enumerated relations.

The rest of the work is organised in subsequently given order. The introduction is followed by the related works section that in greater detail discusses characteristics of the alternative frame-based algorithms used in subsequence detection. Further the experiment is described along with the complete setup and discussion on structure of the used text corpora and its results. Next section focuses shortly on the future work and the publication is concluded with the summary section.

2 Related Work

The core of the work presented in this article is dependent on the SHAPD algorithm that allows for a robust and a resilient computation of a longest common subsequence shared by one or many input documents. SHAPED processes documents by dividing them into a stream of sentences, where unnaturally long sentences (enumerations, itemizations, etc.) are handled by a special procedure [6].

The process is driven by a modular additive hashing function with collision lists. Every term in a sentence is hashed by assigning a number from a previously defined range (during the experiments the limit was set to a large prime number). Further, the individual hashes are summed to represent a sentence. Thanks to the additive nature of hashing function, sentences with changed term order are treated as equivalents. The probability of a collision of sentence hashes, where the individual terms are assigned natural numbers less than several millions, with the average sentence length of 14 is negligible.

The task of matching a longest common subsequence is an important one in many subdomains of Computer Science. Its most naive implementation was deemed to have a time complexity of $O(m_1 * m_2)$ (where m_1 and m_2 are the numbers of terms in compared documents). The question whether it is possible to achieve significantly better results was stated first by Knuth in [9]. First affirmative answer was given in [16] with time complexity $O((m_1 * m_2) / \log(m_2))$ for case when $m < n$ and they pertain to a limited sequence. Another special case with an affirmative answer was given in [12] with time complexity of $O((m_1 + m_2) * \log(m_1 + m_2))$. As to be detailed later, the presented algorithm is another example of a special case where the time complexity is lower than quadratic.

One of the most important implementations of a search for the longest common subsequence is to be found in [13]. This work presents an application of Smith-Waterman algorithm for matching a longest common subsequence in textual data. This is a top achievement of algorithms that do not operate with text frames and their hashes. Other works such as [10], [14] or [17] prove that better efficiency is yielded rather by careful engineering strategies than a fundamental change in time complexity. All of the above cited works use algorithms which time complexity is near quadratic which results in drastic drop of efficiency when dealing with documents of considerable length.

It was first observed in [15] that introduction of special structure that was later referenced to as a shingling (a continuous sequence of tokens in a document) can substantially improve the efficiency of deciding on the level of similarity of two documents by observing a number of common shinglings. Following works such as [4, 3] introduce further extensions to the original idea. A number of works represented by publications such as [8] or [1] provided plausible methods to further boost measuring of the similarity between entities.

The important distinction between those given above and SHAPD is the emphasize on a sentence as the basic structure for comparison of documents and a starting point of determining a longest common subsequence. Thanks to such an assumption, SHAPD provides better results in terms of time needed to compute the results. Moreover, its functioning does not end at the stage of establishing that two or more documents overlap. It readily delivers data on which sequences overlap, the length of the overlapping and it does so even when the sequences are locally discontinued. The capability to perform these makes it a method that can be naturally chosen in plagiarism detection. In addition, it implements the construction of hashes representing the sentence in an additive manner, thus word order is not an issue while comparing documents. The comparison of performance between SHAPD and w-shingling algorithm is given in figure 1. The approach used in w-shingling algorithm performs significantly worse when the task is to give a length of a long common subsequence. Due to the fixed frame orientation, when undergoing such operating w-shingling behaves in a fashion similar to the Smith-Waterman algorithm resulting in a significant drop of efficiency.

The importance of plagiarism detection is recognized in many publications. One might argue that, it is an essential task in times, where access to information is nearly unrestricted and culture for sharing without attribution is a recognized problem (see [19] and [5]). Yet, as this work presents a special case of an algorithm for a longest common subsequence can be used in other applications.

3 Establishing Document Relations through SHAPD

As mentioned above SHAPD has some unique features making it an ideal candidate in in plagiarism related tasks. One of them is the establishment of the most influential documents in a corpus of documents. This can be achieved by a number of various methods, some of which do not analyse the textual content of a document and work only with already prepared relations. The importance of applying of SHAPD is that it works with raw textual data and can detect one or more of the enlisted in the introduction relations. As also mentioned, SHAPD is resilient to a number of problems such as word order in compared sentences and local discontinuities. This greatly diminishes any preparatory work on the tested corpora.

Equipped with such an algorithm, authors begun testing whether it is possible to detect the most influential documents thanks to the enlisted earlier relations.

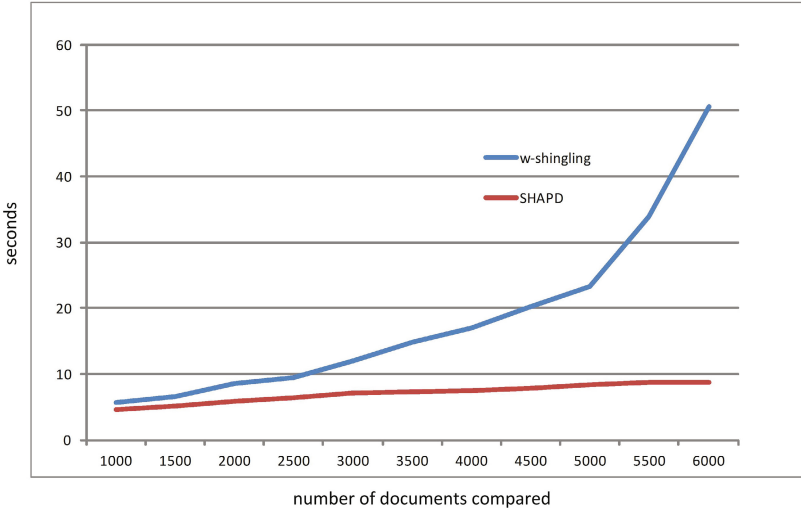


Fig. 1 Results of a comparison of the implementation of SHAPD with the w-shingling given in [4]. The comparison takes a corpus of 3000 documents and compares it to another corpus which size grows from a thousand documents to six thousands. Every second the algorithm performs nearly 1 million of document-document comparisons. The comparison was performed on consumer class notebook equipped with one 8 core Intel processor and 8 GB of RAM. The results are an average of 11 consequent test iterations.

3.1 Experiment Outline

In order to accomplish this a sufficient number of documents had to be assembled. Authors decided to prepare two corpora, one of them being a archive of proceedings from the TREC conference¹ and the other ensemble of scientific publications devoted to various aspects of Computer Science and Information Technology. The TREC corpus covered over a thousand of works from the last decade of the conference. The ensemble corpus was based on over 36 thousands various documents obtained from the CiteSeerX².

The TREC corpus proved to be a coherent one with many references to previously, therein, published works. The CiteSeerX corpus was a heterogenic one with documents from many different subdomains. Therefore in order to prepare meaningful experiments it was partitioned into subcorpora devoted to a particular domain. The partitioning took into account similarity of the documents and labels were provided afterwards.

The partitioning took into account similarity of the documents and labels were provided afterwards. Out of the total of seven different subcorpora three labeled as clustering, ontologies and theory of computing were chosen for further analysis.

¹ <http://trec.nist.gov/proceedings/proceedings.html>

² <http://citeseerx.ist.psu.edu/index>

The process of relations detection was fully automated thanks to preparation of routines that were able to partition each document into three distinct segments. These segments covered the title of the work along with its authors, the main body of the work and the reference section.

Thanks to previously discussed trait of the algorithm sentence awareness, it matched titles with references perfectly avoiding problems with different reference formats. In addition the whole document main body was compared across the available corpus to discover any plagiarism incidents and self-citations. In addition, please note that in order to provide further optimization stop list was applied in order to remove terms of low significance.

One of the relations enumerated in the introduction was commonly used phrases. These were assembled as a type of a stop list. Along with the discovery that they occur at random as a subsequences common for two or more documents they had to be marked by hand by authors to allow for results refinement. The following lists presents a number examples of such common phrases: *permission copy fee material granted provided copies made distributed direct, terms design reliability performance measurement, a variant of the algorithm that uses, immediately following the array referenced by, the word following the array referenced by, please note that in order to provide.*

The overall results of the application of the algorithm in the TREC corpus and the previously discussed three subcorpora demonstrated that the most important relation is the number of citations of a given document and references to other documents from the subcorpus. This is fact is not a novel, especially as there is a number of specialised services that provide detailed statistics on number of citations to any indexed document. Yet, there is no solution, to the best of authors knowledge, that is capable of provision of a list of documents that are best suited to be deemed as the most influential ones for a domain, where as an input a unranked set of documents is given.

Being able to feed a number of documents that share the common topic, one is able to instantly come up with a list of those that are of substantial influence. This is achieved not by artificially built factors but as a result on analysis of the domain itself thus reflecting the beliefs of specialists developing it.

In order to establish a sensible threshold, which document has to pass in order to be enumerated as a member of the influential group, given data set must be analysed to discover what is the total number of subsequences, what is the average of common sequences for documents, what is median and last but not least what documents are mostly referenced.

The quality of assignment to the influential group is strongly dependant on the overall number of processed documents. In authors' opinion, the minimum for any meaningful action is one thousand documents. The statistics computed for the TREC corpus and the three subcorpora are given in table [II](#).

One might argue that when some important articles are absent from the processed corpus one shall obtain distorted picture of the overall domain. As mentioned, without sufficient number of documents that might be true, but beyond certain number even when the most important documents happen to be not included, they shall be

present in references of the documents ranked as fit to become a member of the influential group. Obviously, this can be discovered not only by manual analysis but also by simple routine checking what sequences are shared by documents yet do not lead to a document in the processed data set.

3.2 *Experiment Description*

In order to explain the data presented in table [1](#) a more detailed description has to be given.

The initial experiment on TREC archive allowed to filter out all the documents that have at least two frames in common. This is understood as a situation in which one has a common sequence or sentence when SHAPD is discussed. The overall number of such documents was 322 with a total of 520 common subsequences. Authors had to find an answer on the reason of such situation. It became apparent, after careful analysis that 71.9 % of common frames is the result of a document being referenced or referencing other documents (where actual references are given).

13% of common frames were self-citation cases where no explicit reference were given by authors. 5% of common frames were commonly repeated phrases such as those referenced above. Therefore, the unaddressed common frames can be explained as either:

- an act of plagiarism or,
- documents A and B reference document C which is absent from analysed corpus.

All these accounted for circa 10% of cases.

If one was to remove all the common frames that were identified as self-citation without a reference and those that were identified as commonly used phrases, it is visible that 90.4% of common frames is a result of either being referenced or referencing other documents.

In this group 77.4% of common frames is attributed to being reference where a reminder is a result of referencing. What is more, after sorting documents from the analyzed data set, where sort criterion was a number of documents with which a given document had common frames, observing first tercile of the sorted documents (sort was performed to provide documents in descending order), following facts were noted:

- referencing or being referenced is the source of 93 % of common frames,
- 78.4% of the common frames is explained by being referenced,
- documents from in the first tercile are more referenced among other high ranked documents, reinforcing hypothesis of high influence of these on the whole domain captured in examined data set.

Having established the above, a special crawler was built in order to gather more experimental data from the CiteSeerX web site. As mentioned earlier, over 36 thousand documents were gathered as resources for the described experiments.

First experiment on the gathered resources demonstrated that only 40.7% of the gathered documents had common frames that could be explained by relation of referencing or being referenced. In comparison to the TREC corpus the gathered documents were not coherent in terms of research domain.

Therefore, whole corpus was categorised with fuzzy clustering c-means [2] where Ward's distance was used as a similarity measure. It was used due to the fact that it is characterised as providing clusters with low intra cluster variation and high variation among different clusters [18].

As mentioned in previous subsection, after partitioning the corpus into 7 distinct subcorpora, three of them were examined in terms of search of common frames. The details of experiment are given in table 1 and they follow the explanation given for the TREC corpus. One may see that the results were comparable to the ones for the TREC corpus, thus proving merit of the method. All three subcorpora have clearly defined influential group of documents.

Having performed all the above mentioned operations list of most influential documents for the ontology data set is started with the following documents (only titles are given):

- Understanding and Evolving the ML Module System
- Constructing Flexible Dynamic Belief Networks from First-Order Probabilistic Knowledge Bases
- Reasoning With Conditional Ceteris Paribus Preference Statements
- Global Optimization Using Embedded Graphs
- A Constraint-Based Approach to Preference Elicitation and Decision Making
- ACME: An Architecture Description Interchange Language
- Context-Specific Independence in Bayesian Networks
- Designing Behaviors for Information Agents

4 Future Work

Authors plan to implement further modifications and optimizations so it would be possible to apply SHAPD in a number of new tasks. One of the most important research directions is combining of Semantic Compression [7] and SHAPD so it is possible to overcome another important obstacle in determining whether compared documents contain sequences of similar meaning, yet when different terms are used. This obstacle is usually due to a use of synonyms and hypernyms or hyponyms in place of the originally terms. This is a very frequent practice in real world scenarios thus author believe that such a scenario must be addressed in their work.

In order to test available techniques a number of ready made corpora is available. They will be used as a testbed for future experiments. The priority will be given to the detection level of similar documents, yet speed shall be conserved to the furthest possible degree so that time needed to compute a result could be reasonably short.

Table 1 Results

	TREC	Clustering	Ontology	Computing	Average ³
Number of documents in each corpus/-cluster	1065	1000	1000	1000	
Number of documents with common subsequences with other documents	322	523	453	362	415
Number of common subsequences (original subcorpus)	520	2903	1093	1904	
Self-citation without explicit reference (%)	68 13.08%	308 10.61%	56 16.55%	94 13.71%	13.49%
Commonly used phrases %	26 5.00%	155 5.34%	181 5.12%	261 4.94%	5.10%
Number of common subsequences without self-citation and commonly used phrases (refined subcorpus)	426	2440	856	1549	
Number of referencing documents %	87 20.42%	234 9.59%	88 10.28%	240 15.49%	
Number of referenced %	287	1507 61.76%	519 60.63%	949 61.27%	
Referenced + referencing in refined subcorpus	90.38%	71.35%	70.91%	76.76%	77.35%
Referenced	77.40%	86.56%	85.50%	79.81%	82.32%
First tercile data					
Referenced + referencing in refined subcorpus	93.00%	75.30%	73.09%	80.77%	80.54%
Referenced	78.24%	87.92%	84.34%	77.72%	82.06%

5 Summary

The presented experiment's results allow for a conclusion that detecting the most influential documents in a corpus is possible. It was achieved by the application of SHAPD. It performs better than the most known alternatives and it is capable of much more due to briefly discussed characteristics.

Experiments prove that SHAPD allowed for automated discovery of intra document relations and automated discovery of a set of the most influential documents in analysed domains. The novelty stems from the fact that this analysis is driven only by the input data and no external indicators of importance for any given document. Majority of the procedure to obtain the most influential documents is, as mentioned, fully automated. The only manual labor might be associated with description of a structure of document in order to correctly point to the most important segments and with extending the list of commonly repeated phrases.

The positive results of both the algorithm and its application to selecting the most influential documents in some given domain, shall enable authors to further research various applications and extensions of the presented algorithm. It is found interesting to apply some of the already available indicators in order to further perfect the selection of the most influential documents in given domain.

References

1. Hamid, O.A., Behzadi, B., Christoph, S., Henzinger, M.: Detecting the origin of text segments efficiently. In: Proceedings of the 18th International Conference on World Wide Web, WWW 2009, vol. 7(3), pp. 61–70 (2009)
2. Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag, A.A., Moriarty, T.: A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data. *IEEE Transactions on Medical Imaging* 21(3), 193–199 (2002)
3. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM* 51(1), 117–122 (2008)
4. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G.: Syntactic clustering of the web. *Comput. Netw. ISDN Syst.* 29(8-13), 1157–1166 (1997)
5. Burrows, S., Tahaghoghi, S.M.M., Zobel, J.: Efficient plagiarism detection for large code repositories. *Software: Practice and Experience* 37(2), 151–175 (2007)
6. Ceglarek, D., Haniewicz, K.: Fast Plagiarism Detection by Sentence Hashing. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2012, Part II. LNCS*, vol. 7268, pp. 30–37. Springer, Heidelberg (2012)
7. Ceglarek, D., Haniewicz, K., Rutkowski, W.: Semantic Compression for Specialised Information Retrieval Systems. In: Nguyen, N.T., Katarzyniak, R., Chen, S.-M. (eds.) *Advances in Intelligent Information and Database Systems. SCI*, vol. 283, pp. 111–121. Springer, Heidelberg (2010)
8. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, STOC 2002, pp. 380–388. ACM, New York (2002)
9. Chvatal, V., Klarner, D.A., Knuth, D.E.: Selected combinatorial research problems. Technical report, Stanford, CA, USA (1972)
10. Grozea, C., Gehl, C., Popescu, M.: Encoplot: Pairwise sequence matching in linear time applied to plagiarism detection. *Time*, 10–18 (2009)
11. Hirsch, J.E.: An index to quantify an individuals scientific research output. *Proceedings of the National Academy of Sciences of the United States of America* 102(46), 16569–16572 (2005)
12. Hunt, J.W., Szymanski, T.G.: A fast algorithm for computing longest common subsequences. *Commun. ACM* 20, 350–353 (1977)
13. Irving, R.W.: Plagiarism and collusion detection using the smith-waterman algorithm. Technical report, University of Glasgow, Department of Computing Science (2004)
14. Lukashenko, R., Graudina, V., Grundspenkis, J.: Computer-based plagiarism detection methods and tools: an overview. In: Proceedings of the 2007 International Conference on Computer Systems and Technologies, *CompSysTech 2007*, pp. 40:1–40:6. ACM, New York (2007)
15. Manber, U.: Finding similar files in a large file system. In: Proceedings of the USENIX Winter 1994 Technical Conference, WTEC 1994, p. 2. USENIX Association, Berkeley (1994)
16. Masek, W.J., Paterson, M.S.: A faster algorithm computing string edit distances. *Journal of Computer and System Sciences* 20(1), 18–31 (1980)
17. Mozgovoy, M., Karakovskiy, S., Klyuev, V.: Fast and reliable plagiarism detection system. In: 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, *FIE 2007*, pp. S4H-11–S4H-14 (October 2007)
18. Nock, R., Nielsen, F.: On weighting clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(8), 1223–1235 (2006)
19. Ota, T., Masuyama, S.: Automatic plagiarism detection among term papers. In: Proceedings of the 3rd International Universal Communication Symposium, *IUCS 2009*, pp. 395–399. ACM, New York (2009)

Approximation Algorithms for Massive High-Rate Data Streams

Alfredo Cuzzocrea

Abstract. This paper complements our line of research on *effectively and efficiently processing massive high-rate data streams via intelligent compression techniques*. In particular, here we provide *approximation algorithms* adhering to the so-called *non-linear data stream compression paradigm*. This paradigm demonstrates its feasibility and reliability in the context of emerging data stream applications, such as *environmental sensor networks*.

1 Introduction

The problem of *processing massive high-rate data streams* has already been recognized as a significant one in the wide spectrum of *data stream management and mining research* (e.g., [2, 18, 25, 24, 15, 16, 5]). This kind of data streams are characterized by two relevant aspects that make traditional processing approaches completely inadequate to their effective and efficient processing. A common view to this end is to devise models, algorithms and techniques for *compressing data streams* (e.g., [21, 19, 20, 12, 13, 4]), with the idea that introduced errors are perfectly tolerable for data stream query and mining purposes (e.g., [7]), including higher-level tasks such as OLAP over data streams (e.g., [3, 6, 10]).

Based on this main assumption, in recent years we provide several contributions in the context of effectively and efficiently compressing data streams (e.g., [12, 13, 5, 14, 9, 6, 7, 10, 11]). Among these, in [12] we pose the foundations for *approximate query answering techniques* over data streams that make use of *two-dimensional arrays* for compressing data streams over so-called *time windows*, leading to the definition of *quad-tree-based compressed data structures* named *Quad-Tree Windows (QTW)*. This way, the entire stream is represented in terms of a *B-tree* indexed list of *QTW* called *Multi-Resolution Data stream Summary*

Alfredo Cuzzocrea
ICAR-CNR and University of Calabria, Italy
e-mail: cuzzocrea@si.deis.unical.it

(*MRDS*). *MRDS* is progressively compressed when the available storage space B is not enough to represent new data stream readings, in a *buffer-like manner*. *MRDS* is capable of providing accurate approximate answers to *range-SUM queries* [22]. Following [12], in [9] we make a new contribution where the so-called *non-linear data stream compression paradigm* is proposed. Based on this innovative paradigm, the *MRDS* is not compressed in a “linear” way (like in [12]), but rather in the dependence of *interesting events* that may occur over time. This setting is well-suited for a wide range of emerging data stream applications, such as *environmental sensor networks* (e.g., [14]). The model above conveys in the idea of compressing the *MRDS* in the dependence of a given *degree of approximation* δ , which may vary along the *MRDS*. With respect to the non-linear compression paradigm, in [11] we propose solid theoretical foundations and results, shaped in the form of two theorems and two corollaries on query optimization aspects for data stream query processing deriving from the non-linear compression paradigm. In this paper, we further extend [11] and provide *approximation algorithms* that allow us to compress the *MRDS* according to the non-linear paradigm [11].

The paper is organized as follows. In Section 2, we provide the preliminaries of our research, which are based on results from [11]. In Section 3, we provide the approximation algorithm for compressing *QTW*. In Section 4, we provide the approximation algorithm for compressing *MRDS*. In Section 5, we provide an application scenario of our proposed approximation algorithms. Finally, in 6 we provide conclusions and future work of our research.

2 Preliminaries

In [11], we found a leading relation between the degree of approximation δ and the *QTW* compression process, and the influence of *QTW* leaf nodes and their depth on the accuracy of approximate answers to range-SUM queries, which suggested us to devise a specific, ad-hoc approach that implements the non-linear compression of *QTW*, which, in turn, influences the non-linear compression of the *MRDS*. Here, we recall this critical result [11]. Given a range-SUM query Q and a *QTW*, we introduce the concept of *degree of approximation of Q over the QTW* , denoted by $\delta_{QTW}(Q)$, which is defined as follows:

$$\delta_{QTW}(Q) \approx \Psi \cdot \frac{1}{\varepsilon(Q)} \quad (1)$$

such that Ψ is a function that captures the (inversely proportional) dependency between $\delta_{QTW}(Q)$ and $\varepsilon(Q)$, and $\varepsilon(Q)$ is the *relative approximation error* due to evaluating Q against the *QTW* [11]. Furthermore, given a *QTW* and a population of range-SUM queries \mathcal{Q} , we introduce the concept of *degree of approximation of \mathcal{Q} over the QTW* , denoted by $\delta_{QTW}(\mathcal{Q})$, which is defined as follows:

$$\delta_{QTW}(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \cdot \sum_{k=0}^{|\mathcal{Q}|-1} \delta_{QTW}(Q_k) = \frac{1}{|\mathcal{Q}|} \cdot \sum_{k=0}^{|\mathcal{Q}|-1} \frac{\Psi_k}{\varepsilon(Q_k)} \quad (2)$$

such that: Ψ_k models a function capturing the dependency between $\delta_{QTW}(Q_k)$ and $\varepsilon(Q_k)$ for $Q_k \in \mathcal{Q}$ over the QTW (note that, in the most general case, a *different* Ψ_k for each query Q_k exists), and $|\mathcal{Q}|$ models the cardinality of \mathcal{Q} .

Before introducing the proper non-linear compression approach for QTW , it is necessary to focus the attention on the critical aspect represented by *the issue of checking the current degree of approximation of a QTW* , which, again, is based on Equation (1) and Equation (2).

Given a QTW , in order to check its current degree of approximation, we again exploit Equation (2) but instantiated on an *appropriate* collection of range-SUM queries \mathcal{Q} able to model a “typical” *query-workload* (QWL) against the QTW . This finally allows us to introduce a reliable *query-driven error metrics*, which is founded on the assumption that, fixed the nature and the shape of a typical QWL against the QTW , then any arbitrary query against QTW *probabilistically* follows nature and shape of queries in QWL . Formally, given a QTW and the following three input customizable parameters: ΔS_{Q_l} , ΔT_{Q_l} and σ_l , such that $\Delta S_{Q_l} > 0$, $\Delta T_{Q_l} > 0$ and $\sigma_l > 0$, respectively, a synthetic (range-SUM) query-workload QWL against the QTW is obtained via generating *all* the synthetic queries $Q_{l,k}$ having selectivity equal to $\Delta S_{Q_l} \times \Delta T_{Q_l}$ and overlapping *all* the two-dimensional ranges associated to *internal nodes* of the QTW , such that the following constraints are satisfied: (i) for each query $Q_{l,k} \in QWL$, the two-dimensional range associated to $Q_{l,k}$ is contained by the two-dimensional range associated to the QTW , i.e. $\langle \mathcal{R}_{S,Q_{l,k}}, \mathcal{R}_{T,Q_{l,k}} \rangle \subseteq \langle \mathcal{R}_{S,QTW}, \mathcal{R}_{T,QTW} \rangle$; (ii) for each query $Q_{l,k} \in QWL$, $Q_{l,k}$ overlaps the associated QTW (internal) node n_h by an overlapping region that is *at least* equal to the $\sigma_l\%$ of the volume of the two-dimensional range associated to n_h , i.e. $|\langle \mathcal{R}_{S,Q_{l,k}}, \mathcal{R}_{T,Q_{l,k}} \rangle \cap \langle \mathcal{R}_{S,n_h}, \mathcal{R}_{T,n_h} \rangle| \geq |\langle \mathcal{R}_{S,n_h}, \mathcal{R}_{T,n_h} \rangle| \times \frac{\sigma_l}{100}$. As specifically regards the generation of queries in QWL , this can be easily obtained via ranging ΔS_{Q_l} and ΔT_{Q_l} along the ranges ΔS_{QTW} and ΔT_{QTW} of QTW (see Section (12)), respectively.

Given a QWT and a QWL adhering-to the definition and satisfying the constraints above, we introduce the concept of degree of approximation of QWL over the QTW , denoted by $\delta_{QTW}(QWL)$, via directly extending Equation (2) as follows:

$$\delta_{QTW}(QWL) = \frac{1}{|QWL|} \cdot \sum_{k=0}^{|QWL|-1} \delta_{QTW}(Q_{l,k}) = \frac{1}{|QWL|} \cdot \sum_{k=0}^{|QWL|-1} \frac{\Psi_k}{\varepsilon(Q_{l,k})} \quad (3)$$

such that: (i) $\delta_{QTW}(Q_{l,k})$ models the degree of approximation of $Q_{l,k} \in QWL$ over the QTW (Equation (1)); (ii) $\varepsilon(Q_{l,k})$ models the relative approximation error due to evaluating $Q_{l,k}$ against the QTW (11); (iii) Ψ_k models a function capturing the dependency between $\delta_{QTW}(Q_{l,k})$ and $\varepsilon(Q_{l,k})$ for $Q_{l,k} \in QWL$ over the QTW ; (iv) $|QWL|$ models the cardinality of QWL . Without going into details, it clearly follows that, in the query-driven error metrics model proposed above, by ranging the input

parameters ΔS_{Q_i} , ΔT_{Q_i} and σ_i , it is possible to obtain more or less “rich” QWL , hence more or less accurate estimates for $\delta_{QTW}(QWL)$.

3 Compressing QTW via the Non-linear Paradigm

Before showing how the non-linear compression process works on the whole $MRDS$, in this Section we focus the attention on how a singleton QTW is compressed according to the same paradigm, being the compression of QTW a basic task of the whole $MRDS$ compression process. As we will illustrate in this Section, the tree-based nature of QTW combined with the multi-resolution nature of OLAP queries (including range-SUM queries) offer a meaningful and intuitive way of modeling a partial as well as a full compression of a given QTW in dependence on the degree of approximation δ and the storage space B' needed to represent new arrivals (see Section [11](#)).

Given a QTW to be compressed in dependence on a degree of approximation δ_k and the storage space B' to be released, and the other above-introduced parameters, namely ΔS_{Q_i} , ΔT_{Q_i} and σ_i , the non-linear compression approach of QTW works as follows. First, a syntectic QWL is built from QTW , ΔS_{Q_i} , ΔT_{Q_i} and σ_i , and the degree of approximation of QWL over the (*current*) QTW , $\delta_{QTW}(QWL)$, is obtained. Then, $\delta_{QTW}(QWL)$ is compared with the required δ_k . If $\delta_{QTW}(QWL) \leq \delta_k$, then the non-linear compression algorithm ends as the QTW cannot be compressed while ensuring a (final) degree of approximation which is at least equal to δ_k , under the settings defined by the *actual* model parameters ΔS_{Q_i} , ΔT_{Q_i} and σ_i . In this case, the QTW compression task does not release any amount of storage space, denoted by B'' , to be used to represent new arrivals. Contrary to the latter case, if $\delta_{QTW}(QWL) > \delta_k$, then the non-linear compression algorithm can execute. On the basis of δ_k , the following cases can arise. If $\delta_k = 100\%$, then the QTW must be maintained uncompressed (in fact, in this case, the condition $\delta_{QTW}(QWL) > \delta_k$ fails at a theoretical level), so that no action is taken. Just like the previous one, in this case the amount of released storage space is equal to zero (i.e., $B'' = 0$). If $\delta_k = 0\%$, then the QTW have to be fully-compressed by reducing it to the sole root node, which stores the aggregate value of all the (erased) leaf nodes of the QTW . In this case, $B'' = space(QTW) - 32 - 4$, such that: (i) $space(QTW)$ denotes the occupancy (in KB) of QTW ; (ii) 32 (KB) is the occupancy of the sole QTW root node; (iii) 4 (KB) is the amount of storage space needed to represent the *structural/link information* related to the sole QTW root node. If $\delta_k \in]0, 100[\%$, then the QTW have to be partially-compressed in dependence on δ_k and B' . The latter one is, obviously, the most interesting case to treat.

Here, we adopt the approach of *progressively pruning nodes of the QTW starting from the the older leaf nodes and, recursively, moving towards nodes of the parent level according to a sort or reverse breadth-first searching strategy* [\[23\]](#), until a degree of approximation at least equal to δ_k is achieved, having released the

required storage space B' . It should be noted that the proposed non-linear compression model is a “natural” extension of the previous linear compression model [12] with the novelty that, in the novel model, the compression is driven by the degree of interestingness of events occurring in the target data stream, beyond to the main requirement of releasing the storage space B' . As a consequence, the following main differences with the previous linear compression model arise: (i) a formal relation between the degree of approximation δ and accuracy-aware approximate query answering properties of Q_{TW} is introduced (Equation (1) and Equation (2)); (ii) the compression process evolves according to a reverse breadth-first searching strategy instead that a reverse depth-first searching strategy like the previous compression model (see [12]).

Looking at technical details, in the case $\delta_{Q_{TW}}(QWL) > \delta_k$ and $\delta_k \in]0, 100[\%$, we adopt the following *incremental approach* that aims at simultaneously accomplishing the requirements related to δ_k and B' . We first prune a set of Q_{TW} nodes, said $\mathcal{L}_{Q_{TW}}$, of cardinality $|\mathcal{L}_{Q_{TW}}| = \rho$, being ρ a model parameter, by starting from the older leaf nodes of the Q_{TW} and exploring the Q_{TW} node space via the reverse breadth-first searching strategy. This originates the partially-compressed Q_{TW} , $\widetilde{Q_{TW}}_\rho$. We then check the condition: $\delta_{\widetilde{Q_{TW}}_\rho}(QWL) > \delta_k$ on $\widetilde{Q_{TW}}_\rho$. If this condition is false, then the compression algorithm ends, and the released space due to this compression task is equal to: $B'' = space(Q_{TW}) - space(\widetilde{Q_{TW}}_\rho)$. In this case, the requirement related to δ_k cannot be accomplished. In addition to this, if $B'' < B'$, the requirement related to B' cannot be accomplished as well. Both requirements are thus not satisfied on the *actual* Q_{TW} locally, but the main requirement related to B' may be still satisfied on the whole *MRDS* globally (here, we pursue a best-effort strategy). Contrary to the latter case, if the condition $\delta_{\widetilde{Q_{TW}}_\rho}(QWL) > \delta_k$ is true, we then check the condition: $B'' \geq B'$. If this condition is true, then the compression algorithm ends, with the achievement of both the two requirements (i.e., the one related to δ_k , and the one related to B'), otherwise the previous compression task is iterated again, until the following *final* condition is achieved:

$$\begin{cases} \delta_{\widetilde{Q_{TW}}_\rho}(QWL) \geq \delta_k \\ B'' \geq B' \end{cases} \quad (4)$$

such that the first constraint of Equation (4) (i.e., $B'' \geq B'$) has a *higher priority* over the second constraint of Equation (4) (i.e., $\delta_{\widetilde{Q_{TW}}_\rho}(QWL) \geq \delta_k$).

Due to massive size and incoming rate of of data streams, the above-illustrated compression algorithm could still introduce excessive computational overheads, so that some optimizations are necessary. Among all *dynamic complexity* aspects related to the proposed algorithm, the following two ones are relevant. The first complexity is due to building the synthetic query-workload QWL , which must be performed at each iteration of the algorithm. Fortunately, the issue above can be handled in a very efficient way, as follows. Let QWL^{j-1} and QWL^j denote the QWL generated at the $(j-1)^{th}$ and $(j)^{th}$ iteration of the algorithm, respectively. Let $\mathcal{L}_{Q_{TW}^j}$ denote the Q_{TW} nodes pruned at the $(j)^{th}$ iteration of the algorithm.

Let $N_{Q_{TW}^{j-1}}$ and $N_{Q_{TW}^j}$ denote the number of nodes of the Q_{TW} at the $(j-1)^{th}$ and $(j)^{th}$ iteration of the algorithm, respectively ($N_{Q_{TW}^{j-1}} < N_{Q_{TW}^j}$). It is easy to understand that QWL^j can be obtained from QWL^{j-1} in a very efficient way (for *each* pair of iterations $j-1$ and j of the algorithm), as follows:

$$QWL^j = QWL^{j-1} - \mathcal{Q}(\mathcal{L}_{Q_{TW}^j}) \quad (5)$$

such that:

$$\mathcal{Q}(\mathcal{L}_{Q_{TW}^j}) = \bigcup_{k=0}^{|\mathcal{L}_{Q_{TW}^j}|-1} \mathcal{Q}(n_{\mathcal{L}_{Q_{TW}^j,k}}) \quad (6)$$

where $\mathcal{Q}(n_{\mathcal{L}_{Q_{TW}^j,k}})$ denotes the set of syntectic queries associated to the (internal) Q_{TW} node $n_{\mathcal{L}_{Q_{TW}^j,k}}$ belonging to $\mathcal{L}_{Q_{TW}^j}$. Therefore, it clearly follows that *the dynamic complexity due to generating the synthetic query-workload QWL is linear in the number of iterations of the compression algorithm.*

The second complexity that is relevant to treat is represented by the access cost due to accessing the set of Q_{TW} nodes to be pruned, $\mathcal{L}_{Q_{TW}}$, at each iteration of the compression algorithm. Contrary to the previous case, this complexity cannot be reduced via somewhat optimization. In order to lower the effect of this complexity, the parameter ρ is introduced, such that $|\mathcal{L}_{Q_{TW}}| = \rho$, which determines the number of Q_{TW} nodes to be pruned at each iteration of the algorithm. Obviously, the alternative data access method according to which one singleton node of the Q_{TW} was pruned at each iteration of the algorithm would have introduced an access cost significantly higher than the access cost due to pruning ρ nodes of the Q_{TW} at each iteration of the algorithm. Being ρ a *customizable* input parameter, it can be empirically tuned easily, also in dependence on the particular data stream application scenario considered, in order to introduce low spatio-temporal overheads during the non-linear compression of Q_{TW} . In [10], we provide details and experiments on algorithms for compressing Q_{TW} .

4 Compressing $MRDS$ via the Non-linear Paradigm

The non-linear compression of the $MRDS$ exploits the non-linear compression of Q_{TW} (see Section 3) as a baseline operation. From [11], recall that the event processing layer originates an annotation of the whole temporal dimension of the $MRDS$ by means of tuples of kind: $\langle E_k, t_{E_k,start}, t_{E_k,end}, \delta_k \rangle$, and that the comprehensive non-linear compression process of the $MRDS$ adheres to a best-effort strategy, whose main goal consists in releasing the storage space B' needed to represent new arrivals.

Under the assumptions above, the non-linear compression approach of the $MRDS$ works as follows. First, event annotation tuples are sorted by *descendent values* of δ_k . This sorting phase comes from observing that, given an event annotation tuple

$\langle E_k, t_{E_k, start}, t_{E_k, end}, \delta_k \rangle$, the related portion of aggregate values $\langle [S_0 : S_{N-1}], [t_{E_k, start} : t_{E_k, end}] \rangle$ and the QTW built on top of $\langle [S_0 : S_{N-1}], [t_{E_k, start} : t_{E_k, end}] \rangle$, QTW_{E_k} , a higher value of δ_k means that a higher degree of approximation is required for $\langle [S_0 : S_{N-1}], [t_{E_k, start} : t_{E_k, end}] \rangle$ hence a lower compression of QTW_{E_k} is determined and a lower amount of storage space, denoted by $B''_{QTW_{E_k}}$, can be released from this compression task.

Contrary to this, a lower value of δ_k means that a lower degree of approximation is required for $\langle [S_0 : S_{N-1}], [t_{E_k, start} : t_{E_k, end}] \rangle$ hence a higher compression of QTW_{E_k} is determined and a higher amount of storage space $B''_{QTW_{E_k}}$ can be released from this compression task. On the basis of the amounts of storage space released from portions of aggregate values $\langle [S_0 : S_{N-1}], [t_{E_k, start} : t_{E_k, end}] \rangle$ across the temporal dimension of the $MRDS$, the total amount of storage space released from the whole $MRDS$ due to the non-linear compression process, denoted by B''_{MRDS} , is obtained as follows:

$$B''_{MRDS} = \sum_k B''_{QTW_{E_k}} \quad (7)$$

The non-linear compression process of the $MRDS$ aims at achieving the following final goal, on the basis of a best-effort strategy (see Section 3):

$$B''_{MRDS} \geq B' \quad (8)$$

Equation (8) states that, similarly to δ_k (see Section 3), even B' models indeed a lower-bound parameter for our model. This property of B' is more intuitive than the equivalent one for δ_k , as, for instance, in addition to the amount of storage space $B''_{QTW_{E_k}}$ released from compressing QTW_{E_k} , structural/link information of QTW_{E_k} , as well as of the whole $MRDS$, could be removed during the compression of QTW_{E_k} , thus releasing *additional* amounts of storage space that, like the amounts $B''_{QTW_{E_k}}$, contribute to B''_{MRDS} . Overall, every $MRDS$ compression process can possibly generate an amount of storage space that exceeds the required one, B' , denoted by B^E_{MRDS} . B^E_{MRDS} is defined as follows:

$$B^E_{MRDS} = B''_{MRDS} - B' \quad (9)$$

B^E_{MRDS} can be used to finally obtain a finer-detailed representation of aggregate information stored in the $MRDS$, for instance the one associated to portions of aggregate values related to events of particular relevance for the actual data stream application scenario.

Second, having sorted the event annotation tuples by descendent values of δ_k , we apply the proper best-effort strategy introduced in this research via *compressing the portions of aggregate values stored in the MRDS starting from those having high values of δ_k and moving towards those having low values of δ_k* , until the storage space B' needed to represent new arrivals is released completely.

As regards more practical issues, given an event annotation tuple denoted by $\langle E_k, t_{E_k, start}, t_{E_k, end}, \delta_k \rangle$, the temporal ranges of an *arbitrary* number of QTW

embedded in the *MRDS* can be contained within the time interval $[t_{E_k,start} : t_{E_k,end}]$, or, alternatively, overlap $[t_{E_k,start} : t_{E_k,end}]$. Therefore, in order to detect the set of *QTW* to be compressed in the compression task triggered by the event E_k , we make use of conventional containment/overlap relations among temporal ranges of *QTW* of the *MRDS* and $[t_{E_k,start} : t_{E_k,end}]$. After that, we run the algorithm for compressing *QTW* (see Section 3) on each *QTW* belonging to the so-selected *QTW* set above. In [10], we provide details and experiments on algorithms for compressing the *MRDS*.

5 Application Scenario: Compressing Distributed OLAP Data Cubes in Streaming Settings

Figure 1 shows a possible application scenario of the proposed non-linear data stream compression algorithms, focusing on the context of *Data Stream Management Systems* (DSMS) (e.g., [1]). In particular, we refer the application scenario where buffered *distributed* stream sources populate target *distributed OLAP data*

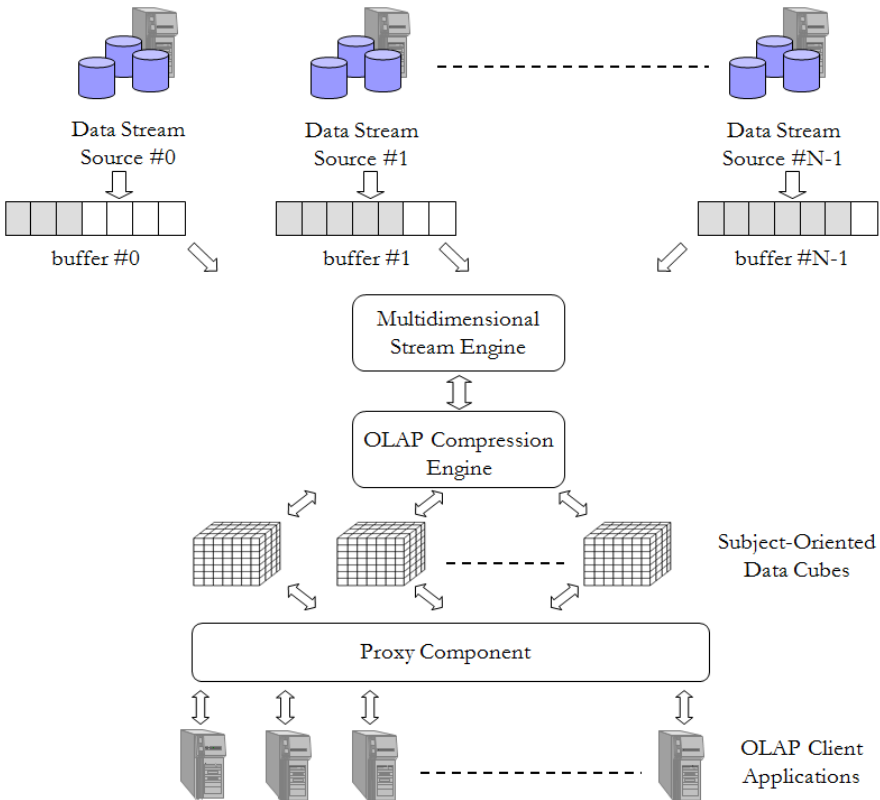


Fig. 1 Application scenario: distributed OLAP data cubes in streaming settings

cubes periodically, and client applications access, query and mine continuous information by means of even complex analytical engines. Here, it is mandatory to introduce compression in order to achieve effectiveness and efficiency during the activities above. In the depicted application scenario, the proposed non-linear compression algorithms are implemented within the so-called *OLAP Compression Engine* (see Figure 1), and are able to produce in output so-called distributed *Subject-Oriented Data Cubes*, meant for *Business Intelligence* (BI) purposes, but still keeping the size-reduction feature at a tolerable query error, and the dependence on interesting (BI) events, according to the guidelines illustrated in the previous Sections.

The distributed nature of the investigated application scenario opens to the challenging issue of how to apply proposed non-linear compression algorithms in a distributed setting. Basically, here two alternatives can be individuated. The first one consists in generating a suitable *MRDS* for *each one* of the target stream source, hence keeping the same approach described in the previous Sections for each stream source. Unfortunately, this approach would convey in scalability issues when the number of stream source grows exponentially. The second alternative is represented by the more appealing *distributed and multiple data stream processing* [17] approach, which pursues the idea of devising models and algorithms capable of processing multiple data streams from heterogenous sources, mostly by means of *computation sharing paradigms* (e.g., [8]). According to these paradigms, our proposed compression approach could be extended as to fit the challenging case of managing distributed and multiple data streams by devising a novel version of the *MRDS* that should be capable of storing the compressed readings of heterogenous data streams, and optimizing the whole in-memory-representation where possible via sharing the compressed representation model. We believe that the latter alternative is the most promising to this end, and it opens the door to exciting future work.

6 Conclusions and Future Work

In this paper, we have complemented our line of research on effectively and efficiently processing massive high-rate data streams via intelligent compression techniques, via providing approximation algorithms adhering to the so-called non-linear data stream compression paradigm. Future work is mainly oriented towards the extending the proposed algorithms in the more probing applicative setting represented by distributed and multiple data streams [17].

References

1. Abadi, D.J., Lindner, W., Madden, S., Schuler, J.: An integration framework for sensor networks and data stream management systems. In: VLDB, pp. 1361–1364 (2004)
2. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and Issues in Data Stream Systems. In: ACM PODS (2002)

3. Cai, Y.D., Clutterx, D., Papex, G., Han, J., Welgex, M., Auvilx, L.: MAIDS: Mining Alarming Incidents from Data Streams. In: ACM SIGMOD (2004)
4. Cormode, G., Garofalakis, M.N.: Sketching probabilistic data streams. In: SIGMOD Conference, pp. 281–292 (2007)
5. Cuzzocrea, A.: Synopsis Data Structures for Representing, Querying, and Mining Data Streams. In: Ferragine, V.E., Doorn, J.H., Rivero, L.C. (eds.) *Encyclopedia of Database Technologies and Applications* (2008)
6. Cuzzocrea, A.: CAMS: OLAPing Multidimensional Data Streams Efficiently. In: Pederesen, T.B., Mohania, M.K., Tjoa, A.M. (eds.) *DaWaK 2009*. LNCS, vol. 5691, pp. 48–62. Springer, Heidelberg (2009)
7. Cuzzocrea, A.: *Intelligent Techniques for Warehousing and Mining Sensor Network Data*. IGI Global (2009)
8. Cuzzocrea, A.: A top-down approach for compressing data cubes under the simultaneous evaluation of multiple hierarchical range queries. *J. Intell. Inf. Syst.* 34(3), 305–343 (2010)
9. Cuzzocrea, A., Chakravarthy, S.: Event-Based Compression and Mining of Data Streams. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) *KES 2008, Part II*. LNCS (LNAI), vol. 5178, pp. 670–681. Springer, Heidelberg (2008)
10. Cuzzocrea, A., Chakravarthy, S.: Event-based lossy compression for effective and efficient olap over data streams. *Data Knowl. Eng.* 69(7) (2010)
11. Cuzzocrea, A., Decker, H.: Non-linear Data Stream Compression: Foundations and Theoretical Results. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012, Part I*. LNCS, vol. 7208, pp. 622–634. Springer, Heidelberg (2012)
12. Cuzzocrea, A., Furfaro, F., Masciari, E., Saccà, D., Sirangelo, C.: Approximate Query Answering on Sensor Network Data Streams. In: Stefanidis, A., Nittel, S. (eds.) *GeoSensor Networks* (2004)
13. Cuzzocrea, A., Furfaro, F., Mazzeo, G.M., Saccà, D.: A Grid Framework for Approximate Aggregate Query Answering on Summarized Sensor Network Readings. In: Meersman, R., Tari, Z., Corsaro, A. (eds.) *OTM-WS 2004*. LNCS, vol. 3292, pp. 144–153. Springer, Heidelberg (2004)
14. Cuzzocrea, A., Gabriele, S., Saccà, D.: High-performance data management and efficient aggregate query answering on environmental sensor networks by computational grids. In: *DEXA Workshops*, pp. 359–364 (2008)
15. Domingos, P., Hulten, G.: Mining High-Speed Data Streams. In: *ACM SIGKDD* (2000)
16. Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining Data Streams: A Review. *ACM SIGMOD Record* 34(2) (2005)
17. Garofalakis, M.N.: Distributed data streams. In: *Encyclopedia of Database Systems*, pp. 883–890 (2009)
18. Garofalakis, M.N., Gehrke, J., Rastogi, R.: Querying and mining data streams: you only get one look a tutorial. In: *SIGMOD Conference*, p. 635 (2002)
19. Gilbert, A., Guha, S., Indyk, P., Kotidis, Y., Muthukrishnan, S., Strauss, M.: Fast, Small-Space Algorithms for Approximate Histogram Maintenance. In: *ACM STOC* (2002)
20. Gilbert, A., Kotidis, Y., Muthukrishnan, S., Strauss, M.: One-Pass Wavelet Decompositions of Data Streams. *IEEE Trans. on Knowledge and Data Engineering* 15(3) (2003)
21. Guha, S., Koudas, N., Shim, K.: Data Streams and Histograms. In: *ACM STOC* (2001)
22. Ho, C.-T., Agrawal, R., Megiddo, N., Srikant, R.: Range Queries in OLAP Data Cubes. In: *ACM SIGMOD* (1997)
23. Knuth, D.E.: *The Art of Computer Programming. Sorting and Searching*, vol. 3. Addison-Wesley (1998)
24. Koudas, N., Srivastava, D.: Data stream query processing. In: *ICDE*, p. 1145 (2005)
25. Muthukrishnan, S.: Data Streams: Algorithms and Applications. In: *ACM-SIAM SODA* (2003)

Comparing Block Ensembles for Data Streams with Concept Drift

Magdalena Deckert and Jerzy Stefanowski

Abstract. Three block based ensembles, AWE, BWE and ACE, are considered in the perspective of learning from data streams with concept drift. AWE updates the ensemble after processing each successive block of incoming examples, while the other ensembles are additionally extended by different drift detectors. Experiments show that these extensions improve classification accuracy, in particular for sudden changes occurring within the block, as well as reduce computational costs.

1 Introduction

Learning classifiers from data streams is one of recent challenges in data mining. Processing large streams of continuously incoming data implies new computational requirements concerning limited amount of memory and small processing time. Moreover, learning is done in non-stationary environments, where underlying data distribution changes over time. It causes changes of target concept definition. These changes are known as *concept drift* [5]. They are usually divided into sudden or gradual concept drifts depending on the rate of changes [10].

Static batch classifiers are unable to adapt to concept drifts and their accuracy decreases with time as they are learned on the outdated examples. Several methods were introduced in the last decade to cope with concept drift [5]. They can be divided into two main groups: trigger based and evolving [12]. *Trigger*-based methods use an on-line classifier with a change detector that reveals if a change occurred. If the change is detected the classifier is re-trained [8]. The popular detector is DDM [4]. *Evolving methods* attempt to update a classifier without a direct detection element. Adaptive ensembles are one those methods. In this paper we are particularly interested in *block-based ensembles*, where component classifiers are constructed from sequential-coming blocks (also called data chunks) of training data. When a new

Magdalena Deckert · Jerzy Stefanowski
Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
e-mail: {magdalena.deckert, jerzy.stefanowski}@cs.put.poznan.pl

block is available, a new classifier is built from it and already existing component classifiers are evaluated. The new classifier usually replaces the worst component in the ensemble.

Accuracy Weighted Ensemble (AWE) [11] is nowadays the best representative of those methods. For review of others see e.g. [5]. However, some authors claim that it may not sufficiently well react to concept drift appearing inside the data block [9]. In particular it concerns delayed reaction to the sudden changes. In our opinion, the reaction of block based ensembles to different type changes should be experimentally studied in more detail. It also poses a new question, whether the block based adaptation mechanism could be extended by a hybridization with the direct detection mechanism. Will this hybridization improve evaluation measures? There already exist two solutions that incorporate a detector into a block-based ensemble structure: Adaptive Classifiers Ensemble (ACE) [9] and Batch Weighted Ensemble (BWE) in [6]. However, ACE is a hybrid solution with the on-line detector, while BWE maintains a specialized block based detector. Furthermore, AWE was criticized with respect to high memory and time costs [6]—reducing it could be another research topic.

The main aim of this study is to carry out comparative experiments of these three block ensembles on 9 different data sets representing different types of changes. In particular, we want to study the impact of introducing a drift detector into a structure of block ensembles. We also modify the earlier version of BWE [6] by considering different ways of modifying weights of component classifiers. All classifiers are evaluated on classification accuracy, memory requirements, and processing time.

This paper is organized as follows. The next section presents related works on detecting concept drift and block ensembles. Section 3 and 4 are devoted to the experimental evaluation of classifiers for various types of concept drift. Section 5 concludes this paper.

2 Related Works

We discuss briefly the most related ensembles. For reviews of other approaches see [5, 7, 8, 10, 12].

First, let us shortly present the Drift Detection Method (DDM) [4] as it inspired solutions in BWE and ACE. This detector is used with online classifier which predicts a class label for each example. The true label of the examples is compared with the predicted one. Classification errors are modeled with a Binomial distribution and for each moment it is verified whether the current error falls into the expected bounds of a warning or alarm level. When a warning level is signaled learning examples are stored in a special buffer. If the alarm level is reached, the previously taught classifier is removed and a new classifier is built from buffer examples.

Ensembles of classifiers have a natural ability to process data that arrive in blocks. Unlike online classifiers that can be modified after reading single examples, block-based ensembles process data arriving in sequential blocks of the same

size. Their main idea is to build a new classifier from each incoming block of data and then to use this block to evaluate performance of all components existing in the ensemble i.e. they receive a weight reflecting their performance. Given k —a fixed number of base classifiers, the less accurate one is replaced by the new one. These classifiers are combined using weighted majority voting. For AWE Wang et al. in [11] proposed that each weight w_i of a classifier is estimated with the formula $w_i = MSE_r - MSE_i$, where MSE_r is mean square error of a random classifier and MSE_i is mean square error of the i th classifier. MSE_r can be expressed by $MSE_r = \sum_c p(c) * (1 - p(c))^2$, where $p(c)$ is the 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 example x is an instance of class c . In each iteration k best component classifiers are chosen to form the final ensemble.

According to [11] [3], AWE is sensitive to the defined size of a block. Moreover, due to building a new classifier for every block, AWE is too demanding with respect to the memory and time cost. These were motivations for introducing Batch Weighted Ensemble (BWE) [6]. It incorporates the Batch Drift Detection Method (BDDM) into the AWE inspired structure of the ensemble. Unlike typical drift detectors BDDM operates on blocks of data. For each example in the block, an accuracy of classification and a standard deviation are calculated incrementally. Then, in order to find trends in the accuracy table, a linear regression is used. Its goal is to estimate a trend in the data, without finding an ideal adjustment. In the next step, we test a gradient parameter a . If its value is less than 0, then we assume that some change occurred. In the end we check, whether drift level was obtained. BDDM verifies levels of change using thresholds for warning and drift. This was inspired by the DDM proposed by Gama in [4] and establish alters with sigma rule in standardized normal distribution. For more details on BDDM see [6]. Batch Drift Detection Method is incorporated into an ensemble. BWE operates as follows. In case when the BWE ensemble is empty, a number of base classifiers are build on bootstrap samples created from the recent block. Otherwise BWE launches BDDM to find trend in the current block. If BDDM signals warning level, the weight of every base classifier is computed using formula $w_i = 0.5 * (1 - \frac{e^{6*(x-0.5)} - e^{-6*(x-0.5)}}{e^{6*(x-0.5)} + e^{-6*(x-0.5)}})$. If size k of an ensemble is exceeded, then the component classifier with the lowest weight is removed. Next, BWE builds a new classifier on the current block and adds it to the ensemble with the weight defined as $w' = maxEnsembleSize - \sum w_j$. In this formula the total weight in an ensemble is constant and equals the ensemble size. If detector signals drift, weights of components are calculated as $w_i = 0.5 * (1 - \frac{e^{4*(x-0.25)} - e^{-4*(x-0.25)}}{e^{4*(x-0.25)} + e^{-4*(x-0.25)}})$. Components, whose classification accuracy is lower than random guessing (their accuracy of classification is less than $\frac{1}{|classes|}$), are removed. If all components are removed, then half of the previous elements with the highest weights are restored. The formulas for calculating components' weights have evaluated since the previous version. We changed them in a way that they decrease slower with an increase of classification error. However, after the error of classification reaches a threshold value (0.5 for warning and 0.25 for drift) the weights decrease faster. The weights functions are similar to the reverse logistic function.

Another block ensemble that is incorporated with an explicit on-line drift detector is Adaptive Classifiers Ensemble (ACE) proposed in [9]. This system consists of one online learner, many batch learners and a drift detection mechanism. The drift detector is different from DDM and BDDM in a way that it uses confidence intervals for proportions to define appropriate drift thresholds. The authors define upper endpoint and lower endpoint of confidence intervals for the suitability. If the suitability of the best hypothesis is less than lower endpoint or higher than upper endpoint, the system assumes that concept drift occurred. ACE system operates as following. When a new learning example is available, the on-line classifier is updated and this example is stored in a buffer. If drift detector signals change or the number of examples in the buffer exceed the maximum size, a new batch learner is created using examples from the buffer. According to the author simple learning algorithms are better as a component classifiers. Original ACE does not stop growing, however we introduced maximum size of the ensemble in order to obtain more comparable results. After a new base classifier was constructed, the online learner and the memory buffer are erased. ACE does not retrain the batch learners. Predictions of an online learner and component classifiers are combined using weighted majority voting—although a different one than in AWE, see [9].

3 Experiments Setup

We chose 6 different classifiers for experimental comparison. First 3 are different block-based ensembles AWE, BWE with BDDM and ACE. As we are interested in studying the role of drift detectors we additionally constructed a batch version of Gama’s DDM, which alerts after the block of passing examples. In comparison to BDDM, it does not contain a trend regression element, so we can evaluate its potential impact. As additional 3 classifiers we use BWE with this batch version of DDM, and two single classifiers: a decision tree (WEKA J48) with standard DDM and the same tree classifier with the batch version of DDM. They were chosen to compare ensembles versus simple classifiers also accompanied with change detectors.

Nearly all classifiers, in particular (AWE and BWE), were implemented in Java and were embedded into the Massive Online Analysis framework for mining streams of data. More about the MOA project can be found in [1] and at the website¹. All component classifiers were constructed using the C4.5 tree (J48 from WEKA)—to be consistent with related works [11,9]. In order to obtain a more precise description of the current block we used unpruned trees. We wanted the component classifiers to reflect only knowledge obtained from one block of data. Thanks to this they will be more specialized for different knowledge regions. We also tested Naive Bayes as a base component. However, the tendencies in the obtained results are similar to the ones for decision trees, so because of the page limit we confine ourselves only to the C4.5. ACE ensemble was used in another implementation originally provided by

¹ See: <http://moa.cs.waikato.ac.nz/>

Nishida, which was run by a proxy in MOA². Single classifiers were WEKA classes combined with MOA framework.

Taking the block perspective we chose three different sizes: 500, 1000, and 1500 examples—inspired by an earlier, related research [11]. To estimate classification performance we used the EvaluateInterleavedTestThanTrain method from MOA. It first use each example in the stream to assess a classification accuracy and than this example can be used to re-train/update the classifier. Evaluation measures were recorded after every 10, 50, or 100 examples. Besides the total classification accuracy we also evaluate values of accumulated processing time from the beginning of the learning phase and the size of current model (expressed by memory size).

We looked for several datasets involving different types of changes, such as gradual drifts, sudden drifts, blips (representing rare events—outliers in a stable period, which a good classifier should not treat as real drifts), stability periods (no drifts for which a classifier should not be updated) and complex/mixed changes. We use 9 different datasets representing: 3 real datasets (often considered in related works) and 6 artificial datasets obtained with MOA generators. Detailed characteristics of these datasets are given in Table 1.

Table 1 Characteristics of datasets

Dataset	Examples	Attributes	Classes	Change type	Properties
CovType	581012	54	7	unknown	N/A
Electricity	45312	8	2	unknown	N/A
Poker	829201	11	10	unknown	N/A
Hyperplane	100000	10	4	gradual	slow magnitude of change $t=0.001$
RBFGradual	100000	20	4	gradual	parameters: $p=5001$, $a=45$, $w=1000$
STAGGER	100000	3	2	sudden	change - every 3001 examples
RBFsudden	100000	20	4	sudden	change - every 5001 examples
RBFblips	100000	20	4	blips	parameters: $p=24990$, $a=80$, $w=200$
RBFNoDrift	100000	10	2	N/A	default parameters

Electricity is a real data set containing energy prices from the electricity market in the Australian state of New South Wales. Poker is also a real dataset consisting of examples reflecting a hand of five playing cards. CovType contains information about the forest cover type for 30 x 30 meter cells obtained from the US Forest Service. For these real data it is difficult to state what kind of drift occurs. However, we chose them because they are widely used as benchmarks by the concept drift community. Then, for each synthetic dataset, the particular type of the concept drift was introduced. Hyperplane is a popular generator (often considered in the literature on changing environments) that creates streams with gradual concept drifts by rotating the decision boundary for each concept. STAGGER is a data generator that creates streams with sudden concept drifts. Original STAGGER data set is easy to learn. In order to make it more difficult we introduced faster changes with

² We are grateful to Dr. Kyosuke Nishida for providing his implementation of ACE for our experiments.

recurring concepts. RandomRBF generates a random radial basis function stream. We created different versions representing either gradual, sudden drifts. Moreover, we generated special versions with blips and a stable situation with no drifts—see Table 1. Notice, that gradual or sudden changes are started in the early part of blocks to simulate a more difficult situation for block based ensembles. More precise descriptions of used generators with their parameters can be found in MOA Manual [2].

4 Experimental Results

Although we carried out more experiments due to page limits we have to present only the most representative results. This is why we focus on results from one data block size = 1000 examples. Results for other sizes showed similar rankings of classifiers. All compared algorithms were evaluated in terms of classification accuracy, memory usage and total processing time. The accuracy values were averaged over recording time points (every 100 examples, more frequent records did not influence the results). Memory is similarly averaged. The values of all measures on datasets are presented in Tables: 2, 3 and 4. We will interpret them in the next section.

Table 2 Average values of classification accuracy [%]

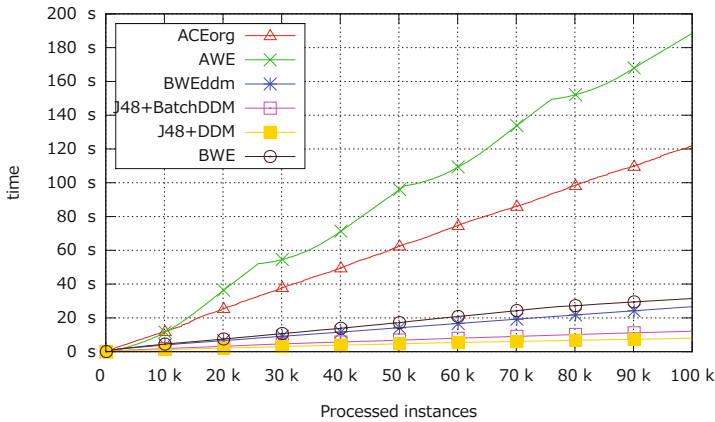
Dataset	ACE	AWE	BWE+BDDM	BWE+DDM	J48+BatchDDM	J48+DDM
CovType	89.38	81.52	82.60	68.43	62.58	42.36
Electricity	87.73	73.53	71.41	67.97	63.21	52.29
Poker	86.48	78.32	75.49	64.09	14.97	46.56
Hyperplane	81.58	70.91	77.11	71.39	70.33	71.13
RBFGradual	85.37	75.25	74.49	55.41	56.25	61.89
STAGGER	98.80	78.30	78.30	56.22	60.28	80.01
RBFSudden	85.49	75.37	74.40	51.74	56.26	59.82
RBFBlips	87.52	88.41	85.55	69.61	65.03	76.75
RBFNoDrift	85.02	88.01	87.41	87.18	81.51	81.51

Table 3 Average amounts of used memory [MB]

Dataset	ACE	AWE	BWE+BDDM	BWE+DDM	J48+BatchDDM	J48+DDM
CovType	0.40	5.49	0.79	1.19	1.74	0.65
Electricity	0.26	0.76	0.58	0.42	0.43	0.07
Poker	0.36	1.48	1.21	1.34	0.64	0.39
Hyperplane	0.35	0.63	1.06	0.70	0.45	0.42
RBFGradual	0.41	1.40	0.42	0.70	0.64	0.44
STAGGER	0.40	0.50	0.07	0.09	0.26	0.22
RBFSudden	0.41	1.40	0.43	0.78	0.64	0.43
RBFBlips	0.40	4.13	0.82	0.72	0.59	0.35
RBFNoDrift	0.35	4.02	0.79	0.48	0.27	0.20

Table 4 Total processing time [s]

Dataset	ACE	AWE	BWE+BDDM	BWE+DDM	J48+BatchDDM	J48+DDM
CovType	350.04	897.01	338.30	380.85	95.78	63.13
Electricity	10.97	20.83	11.23	8.49	6.16	2.28
Poker	176.50	629.35	287.56	306.68	106.33	74.26
Hyperplane	55.47	35.74	37.27	23.38	10.76	10.80
RBFGradual	141.15	68.00	20.34	26.36	12.28	9.52
STAGGER	24.73	33.09	3.96	4.87	8.70	7.94
RBFsudden	144.30	68.50	20.69	28.75	12.03	9.30
RBFblips	122.10	188.64	31.54	26.64	12.12	7.96
RBFNoDrift	46.86	228.14	28.25	17.08	7.21	6.04

**Fig. 1** Processing time for RBFblips data

For better insight into dynamics of learning we prepared figures of these measures after processing every learning example. Again due to the space limits we present only the representative figures for real datasets CovType and Electricity. Then, changes of accuracies are shown for Hyperplane (RBFGradual looks quite similar) and RBFsudden (STAGGER figure represents similar behavior of classifiers)—see Figure 2.

In case of memory changes we give figures for two data sets to show characteristic differences in ensembles performance. For data with gradual drift, BWE uses more memory than usual. This may be the result of many warnings signaled, for which pruning is not active yet. The differences of processing time between algorithms are quite analogous (AWE is the most time consuming, while single classifiers are the fastest ones), so one figure is given.

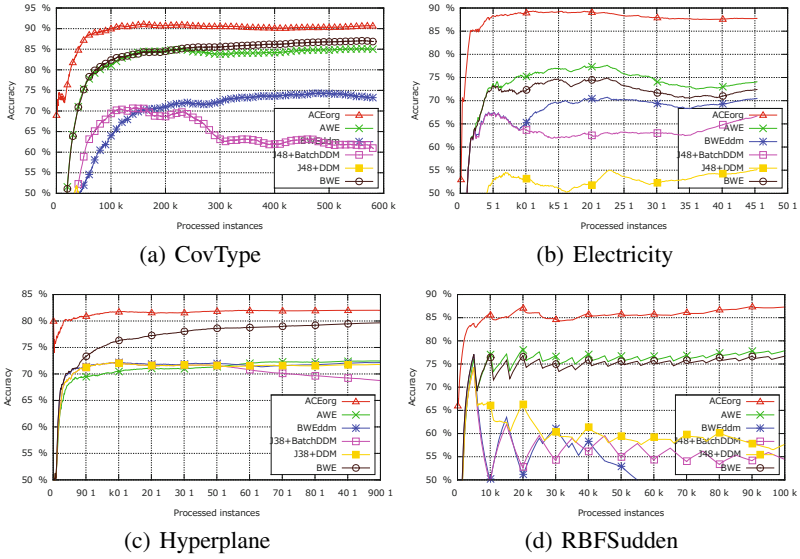


Fig. 2 Classification accuracy for selected datasets

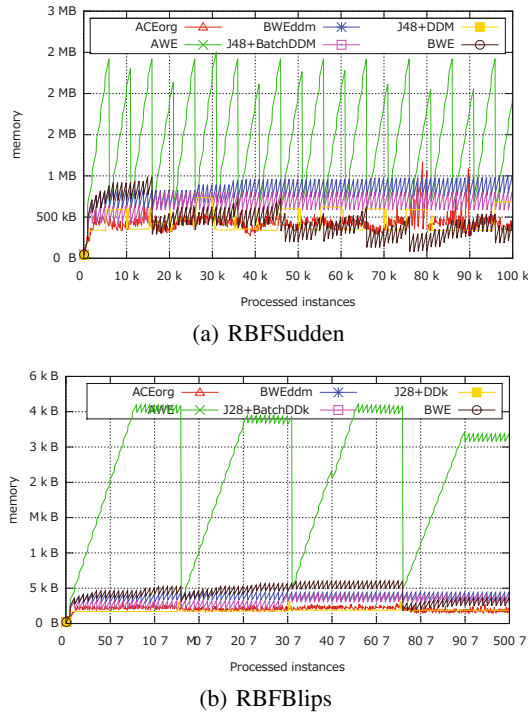


Fig. 3 Changes in memory usage

5 Discussion of Results and Final Remarks

Results of total accuracy show that ACE achieves the highest value of classification accuracy for most of the data. Its superiority is particularly visible for data with sudden changes. Both for STAGGER and RBFSudden it produces the highest increase of over 10%. Its good performance is also reflected by analysing a graphical plot, where its plot is less influenced by fast drift, while in other classifier plots one can clearly see moments of drift occurrence. These "switches" are also clearly visible in the respective figure for changes of memory usage. Moreover, looking at real data figures, one can notice that ACE is able to achieve the best accuracy results much faster than other classifiers.

We can explain its good performance by using drift detection combined with an online learner in a hybrid structure with ensemble components. As many drifts are started in the beginning of blocks, ACE can detect them earlier than block classifiers and update weights of components faster as well as add a new classifier to the ensemble that reflects the recent changes. BDDM's reaction is postponed to the end of each block. It is also interesting to note that using a weighted set of classifiers is always more accurate than the single classifier integrated with DDM. In case of a single tree, classical on-line DDM detector is nearly always better than the batch version.

The only exception to ACE best performances are non typical data sets like RBF-Blips or NoDrift, where BWE and AWE are more accurate. For RBFNoDrift, BWE with BatchDDM also slightly outperforms ACE. Differences of accuracy between BWE and AWE are quite small on real data and these with sudden concept drift (see e.g. their plots for RBFsudden). For all data with gradual concept drift BWE obtains better results than AWE. With the number of processed examples it even achieves a level closer to ACE. The BDDM may signal enough warnings for BWE if changes develop wider inside the block, while it may be insufficient for faster changes, where an on-line detector is superior to start the update of the ensemble. Moreover, considering regression inside, BDDM part was better than incorporating simple batch DDM to BWE.

On the other hand, we can suspect that the number of alerts produced by BDDM for some data can be too small to sufficiently compete with AWE which continuously tries to add new components. In other recent [3] study it was shown that the AWE idea could be outperformed by AUE ensembles where component classifiers are incrementally updated by the recent blocks. This AWE paradigm is too costly. Results clearly show that AWE uses the largest amount of memory. For every block AWE builds a new base classifier, always use the full size of the ensemble and after some period of time all out-dated components may be deleted. This behavior is reflected by characteristic periodic patterns in figure 3. Of course single classifier are the least costly. Then, BWE often uses less memory. In case of sudden changes or no drifts BWE uses 5 times less memory than AWE. In case of real data sets differences range between 1.2 and 7 times depending on the complexity of the data. BDDM in BWE limits number of constructed components (as we checked usually the number of components in BWE is smaller than in AWE). One exception from this rule is visible for slow gradual changes in Hyperplane data set. In this case,

BWE uses a lot of memory that may be a result of a large amount of warning signals detected. When a warning is detected, BWE does not prune the ensemble. This reflects in the increased number of component classifiers (even the maximum size of an ensemble may be achieved). ACE also uses less memory than AWE. However, one should be careful as ACE is an external code run from MOA. Representative results on memory usage are presented in Figure 3.

Results obtained on processing time are similar for every data set. AWE needs the longest period of time. ACE in most of the cases also works for a longer time. BWE, thanks to the fact that it does not build a classifier for every block of data, acts faster than other ensembles. However, for real data sets, which are more complex, ACE operates a similar amount of time to BWE. The fastest ones are single classifiers but this does not compensate results obtained on total accuracy.

To sum up, in our opinion experimental results show the usefulness of incorporating a drift detector inside the block-based structure of adaptive ensembles. While accuracy is the main criterion, an approach with on-line learning is the better choice. However, from the computational costs, BDDM was more efficient. In future research, it could be interesting to construct a hybrid approach taking advantage of both of these solutions.

References

1. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive Online Analysis. *Journal of Machine Learning Research (JMLR)* 11, 1601–1604 (2010)
2. Bifet, A., Kirkby, R.: *Massive Online Analysis Manual*. COSI (2009)
3. Brzeziński, D., Stefanowski, J.: Accuracy Updated Ensemble for Data Streams with Concept Drift. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS (LNAI), vol. 6679, pp. 155–163. Springer, Heidelberg (2011)
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 Publishers (2010)
6. Deckert, M.: Batch Weighted Ensemble for Mining Data Streams with Concept Drift. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 290–299. Springer, Heidelberg (2011)
7. 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)
8. Kuncheva, L.I.: Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. In: *Proc. 2nd Workshop SUEMA 2008 (ECAI 2008)*, Greece, pp. 5–10 (2008)
9. Nishida, K., Yamauchi, K., Omori, T.: ACE: Adaptive Classifiers-Ensemble System for Concept-Drifting Environments. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 176–185. Springer, Heidelberg (2005)
10. Tsymbal, A.: The problem of concept drift: Definitions and related work. Technical Report, Trinity College, Dublin, Ireland (2004)
11. 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)
12. Zliobaite, I.: Learning under Concept Drift: an Overview. Technical Report, Vilnius University, Lithuania (2009)

Adapting Travel Time Estimates to Current Traffic Conditions

Przemysław Gawel, Krzysztof Dembczyński, Robert Susmaga,
Przemysław Wesolek, Piotr Zielniewicz, and Andrzej Jaszkievicz

Abstract. The paper demonstrates drifts in travel time estimates of a floating car data based car navigation system. The operation of such a navigation system starts with collecting floating car data, i.e. multi-channel stream data sent in from moving cars. These dynamic data are then processed in an elaborate, multistage procedure, aimed at estimating the travel time and constituting an essential component of optimal route planning, which can effectively find not only the shortest, but also the fastest road connections, always taking into account the current traffic conditions. The experiments present the ability of the navigation system to detect and handle unusual traffic situations, like unexpected jams caused by sudden road accidents, which manifest themselves in the drifts of travel time estimates. All experiments were conducted on exclusively real-life data, provided by NaviExpert, a Polish car navigation company.

Keywords: floating car data, travel time prediction, concept drift.

1 Introduction

A car navigation system today is something more than the yesterday's route planner, which could find a road connection and navigate one from A to B in unknown surroundings. This fundamental, route finding functionality, typical for earlier versions of the navigation systems, has now been augmented

Przemysław Gawel
NaviExpert Sp. z o. o., Dobrzyckiego 4, 61-692 Poznań, Poland
e-mail: pgawel@naviexpert.pl

Krzysztof Dembczyński · Robert Susmaga · Przemysław Wesolek ·
Piotr Zielniewicz · Andrzej Jaszkievicz
Institute of Computing Science, Poznań University of Technology,
Piotrowo 2, 60-965 Poznań, Poland

with additional features, which make it possible to find routes optimized for personalized criteria, like route length or travel time. In result, the modern system can find, independently, shortest routes and fastest routes, the latter requiring an accurate travel time prediction model. But the travel time depends not only on the type of road (which is a static and easily available piece of information) but also on the current traffic conditions (a very dynamic type of information, not straightforward in collecting and managing). Modern car navigation system can obtain this type of traffic information from the floating car data, i.e. multi-channel stream data sent in from moving cars.

The problem of learning adaptive models from floating car data for the travel time estimation has gained an increasing attention in recent years. The data may be obtained from different sources like loop detectors [9], other stationary sensors [7], or GPS devices [1]. In the former cases, the learning problem is easier, since observations are continuously sampled from all detectors. In the case of GPS data, the observations are sparse and unevenly distributed, however, they often cover a larger part of the road network. If a travel time estimation model is to find the shortest path between any two points of the road traffic network, the extensive coverage of the network is an important aspect of the data.

The presented model, which extends our previous work [4], consists of a static and a dynamic component, which target the long-term and the short-term prediction horizon, respectively. Such architecture of the system is commonly used in similar studies [8, 2, 1, 3]. Using exclusively real-life data, provided by NaviExpert, we show how the travel time estimates change, or drift, in reaction to unusual traffic situations, like unexpected jams caused by sudden road accidents, only to return to their typical values after the situation stabilizes. The model used in this paper is generally able to detect and handle most of those unusual traffic situations.

The rest of the paper is organized as follows. After this introduction, Section 2 describes the intricacies of the complex data processing, while Section 3 states the computational problem that is addressed and solved. Its solution is described in brief in Section 4. Section 5 recounts two experimental studies involving the static and dynamic components of the model. The paper is concluded in its final section.

2 Floating Car Data

The floating car data originate mostly from vehicles equipped with GPS devices. These devices constitute individual data sources, with each of them transmitting continuously (in practice: at some short time intervals) a separate stream of data. The floating car data constitute thereby an essentially amorphous collection of data streams coming from different sources, or subjects, which are stored and appropriately processed. During their processing,

the data undergo several phases of structuralization. In phase one, raw geographical positions of the subjects are converted to passages through basic road units that the underlying road network is logically divided into, i.e. the road segments. This means, first of all, the identification of the road segment on which the subject is currently situated. After obtaining further observations from the same subject, its movement can be identified and the direction of movement determined. This procedure is repeated until the subject terminates the service or moves to another segment of the network. In the end, multiple pieces of information characterizing the subject's positions are converted into one piece of information describing the subject's passage through the segment. This passage information is further supplemented with some auxiliary information, like time and date of the event.

In the second phase, the (dynamic) information on passages through road segments, together with other available forms of (static) information on the segments, serves to build learning data sets, i.e. data sets that are used to learn regression models with the aim of predicting the segment travelling time. Two different types of data sets that can be created at this stage: sets of past observations and sets of recent observations. These two types are, in a sense, complementary; the past observations are often out-of-date, but usually more numerous, while the recent observations are often scarce, but usually more up-to-date. The first type of data sets can be used with models that are constructed basically once, ahead of time (further referred to as static models). Because such models are trained on a relatively large data sets, they exhibit stable predictions (low variance), but they are potentially less reactive to dynamically changing traffic situation (high bias). The second type of data sets can be used with models that are constructed periodically

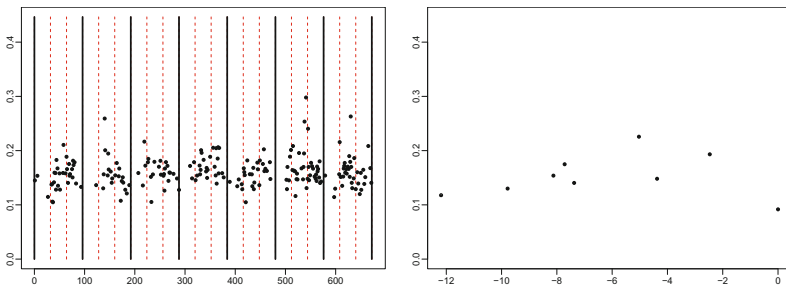


Fig. 1 (Left) Input data for the static model: numerous passage times (in minutes, the y -axis) through a road segment observed in a three-month period (mapped to a one-week interval); the unit of the x -axis corresponds to a time interval of 15 minutes (thus, the values of the x -axis range from 1 to 672; black lines separate days of the week, from Sunday till Saturday; red dotted lines mark 8 a.m. and 4 p.m.). (Right) Input data for the dynamic model: scarce passage times (in minutes, the y -axis) observed in the past 3 hours; the x -axis represents the time before the time point t_0 ; the unit of the x -axis corresponds to a time interval of 15 minutes.

(further referred to as dynamic models). Because these models are trained on relatively small data sets, they exhibit less stable predictions (high variance), but they can potentially react very dynamically to current traffic situations (low bias).

The experiments presented in this paper illustrate a combined application of both types of models. The difference in input data for the models is shown in Figure 1. For the static model, we take into account all available observations (mapped to the time interval of one week), while for the dynamic model we take into account only the most recent observations.

It should be stressed that because the geographical distribution of the original GPS data is generally unpredictable and often uneven, the sizes of the created data sets can be fairly erratic, making all subsequent analyses of these data still more difficult.

3 Problem Statement

The goal of the problem can be stated as a prediction of an unknown value of a vehicle travel time y_{st} on a particular road segment $s \in \{1, \dots, S\}$ in a given time point t . The task is then to find a function $f(s, t)$ that estimates, in the best possible way, the value of y_{st} . The accuracy of a single prediction $\hat{y}_{st} = f(s, t)$ is measured by a loss function $L(y_{st}, \hat{y}_{st})$, which determines the penalty for predicting \hat{y}_{st} when the true value is y_{st} . a reasonable loss function in this case is the squared error loss:

$$L(y_{st}, \hat{y}_{st}) = (y_{st} - \hat{y}_{st})^2.$$

Ideally, we would like to get a model $f(s, t)$ that minimizes the *expected* risk:

$$f(s, t)^* = \arg \min_f R(f) = \arg \min_f E_{(s,t)} E_{y|(s,t)} [(y - f(s, t))^2].$$

Since this is directly impossible, as the distribution of y given (s, t) is hardly ever known, we rely on a set of training samples, $\{(y_i, s_i, t_i)\}_{i=1}^N$, and construct a model that, instead, minimizes the *empirical* risk:

$$R_{emp}(f) = \frac{1}{N} \sum_{i=1}^N L(y^{(i)}, f((s, t)^{(i)})), \quad (1)$$

possibly with a kind of regularization over the function f to prevent overfitting of the model [6].

We additionally assume that for each road segment s and time point t , a vector $\mathbf{x}_{st} = (x_{st1}, x_{st2}, \dots, x_{stn})$ of attributes, which describe the segment and the time point, is known. Without the loss of generality, we assume that attribute values are real numbers, i.e. $\mathbf{x} \in \mathcal{R}^n$.

4 Adaptive Travel Time Prediction System

In this section, we introduce a procedure that tries to fully exploit the nature of the floating car data. The procedure comprises two models. The first model, the static one, is responsible for predicting overall trends in the traffic. It assumes that the traffic undergoes periodic changes, but is otherwise static. The model is created from a set of past observations, discovering (potentially existing in the data) repeatable traffic flow patterns (e.g. “at every Sunday morning, on a road segment in the city center, the traffic is low”). This constitutes its strength (the stability of predictions, ensured by large data samples and the ability to predict for the long-term, e.g. with a horizon of a few days), but also its weakness (the inability to react to dynamically changing, non-periodic traffic conditions). This poor reactivity is also the reason for introducing the second model, the dynamic one, which exploits the most recent, real-time observations with the aim to improve the short-term predictions of the static model. The algorithms for the static and dynamic model are further described in the next two subsections.

4.1 Static Model

Construction of the static model is similar to the typical regression task. To deliver the right prediction, the model uses attributes that describe a given segment at a given time point. The training data are represented in a tabular form $\{(y_i, \mathbf{x}_i)\}_{i=1}^N$. In this study we use rather simple static models that exploit only a limited number of features describing a road segment and a time point. However, the models described here behave fairly satisfactorily and in sake of readability we limit our discussion to them.

The simplest method for predicting the travel time y relies on estimating a single value from all observations. Such a value corresponds to the average unit travel time for the considered road network and the time interval. More precisely, we compute the average inverse velocity (the average travel time for a length unit) over all historical observations:

$$\bar{v}^{-1} = \frac{\sum_{i=1}^N y^{(i)}}{\sum_{i=1}^N x_l^{(i)}},$$

where x_l is the length of the l -th segment. The prediction for a given road segment is then given by:

$$\hat{y}_{gm} = f_s(\mathbf{x}) = x_l \times \bar{v}^{-1}. \quad (2)$$

This form of the prediction is reasonable, as the average inverse velocity is the solution to the optimization problem:

$$\bar{v}^{-1} = \arg \min_a \sum_{i=1}^N x_l^{(i)} \left(\frac{y^{(i)}}{x_l^{(i)}} - a \right)^2,$$

where the length of the segment is multiplied by the loss for single observations. In other words, if we minimize the weighted squared loss on a training set, the average inverse velocity is the best possible choice for the estimated single value. We refer to this model as the *global mean*.

The second method, referred to as the *segment mean*, averages the travel time on each road segment separately. Although more specific than the global mean, it is still primitive enough to ignore the time point of the passage:

$$\hat{y}_{sm} = f_s(\mathbf{x}) = \frac{\sum_{i: x_{id}^{(i)} = x_{id}} y^{(i)}}{\sum_{i: x_{id}^{(i)} = x_{id}} 1}. \quad (3)$$

The third model, referred to as the *segment/time period mean*, additionally considers information about the time point of the passage. Using expert knowledge on weekly traffic trends, we define five time periods and separately compute the mean travel time for each road segment and each time period according to (3).

4.2 Dynamic Model

The goal of the dynamic model is to use the most observations to improve the predictions of the static model $f_s(\mathbf{x}_{st})$ in the short-term. The dynamic model is introduced to account for those the changes in traffic that cannot be explained by exploiting long-term and periodic behaviour.

Dynamic model f_d consists of two parts. The first part is constructed as a time series model for each road segment. Prediction \hat{y}_{st_0} for a given segment s and time point t_0 is computed using previous observations y_{st} , $t < t_0$, from segment s . Training data are represented in a form $(y_{st_1}, y_{st_2}, \dots, y_{st_{N_s}})$, for each segment $s \in \{1, \dots, S\}$, where N_s is the number of observations for s . In this paper we use a simple moving average model that consists in averaging all past observations from a given time interval t :

$$f_d(s, t) = \hat{y}_{st_0} = \frac{\sum_{t_0 - t_i < t} y_{st_i}}{\sum_{t_0 - t_i < t} 1}. \quad (4)$$

The second part of the model takes as input the prediction from the static model $f_s(\mathbf{x}_{st})$, the prediction from the moving average model $f_d(s, t)$, and produces the final travel time estimate as a linear combination of the segment length and the two previous predictions:

$$f(s, t) = a_0 x_l + a_1 f_s(\mathbf{x}_{st}) + a_2 f_d(s, t).$$

This model is thus a kind of a cascade, in which the static prediction is combined with the dynamic moving average. The model is trained by linear regression every 5 minutes on the most recent observations from a time window of a few past hours. The coefficient a_0 models the recent increase or decrease of the average travel time (per unit length); the coefficient a_1 is responsible for the proportional adjustment of the static model to the current traffic, while the coefficient a_2 determines the reliability of the prediction computed from the most recent observations.

5 Experimental Study

In our experiments we use real-life floating car data provided by NaviExpert. All the used data were collected from a pre-defined geographical area in a pre-defined time range.

The area of observations ranges from 16.94190° N to 16.95980° N and from 52.39294° E to 52.41417° E, covering a little less than 3 square kilometers in the city of Poznań (a city in Poland of about half a million population). The area contains two important roundabouts: Rondo Śródka and Rondo Rataje, with Jana Pawła II Street between them, as well as two important bridges: Most Bolesława Chrobrego and Most Św. Rocha. This particular area was chosen because in 2011, on 26th of September, it was affected by a specific incident that we study in greater detail in the described experiment.

The time range of the observations spans four weeks in 2011: from September 12th till October 10th, with the exception of the night hours (from midnight until 5 a.m.).

In total we use four methods for travel time estimation: the global mean (GM), the segment mean (SM), the segment/time period mean (STP), and the dynamic model (DM). For computing linear regression we use the Weka package [5]. To build the dynamic model we take the most recent observations from a time window of exactly one hour.

The next subsections describe two experiments. The first one concerns the general performance of the models, while the second concentrates on the drift of travel time estimates.

5.1 General Performance

The first experiment evaluate the general performance of the models. To this end, split the data into a learning and a testing part: the learning set extends from September 12th till September 25th, while the test set extends from September 26th till October 10th. The static models are built using only the learning set, while the dynamic model additionally uses the most

Table 1 Results of the four models on test sets. Mean absolute (MAE) and root mean squared error (RMSE) are reported.

model	MAE[min]	MAE[%]	RMSE[min]	RMSE[%]
Global Mean	0.3307	119.80	0.8556	108.00
Segment Mean	0.2761	100.00	0.7922	100.00
Segment/Time Period Mean	0.2649	95.97	0.7776	98.15
Dynamic model	0.2556	92.60	0.6415	80.98

recent observations from the test set (but each prediction is entirely based on earlier observations). The performance of all models is presented in Table 1. We report both mean absolute error (MAE) and root mean squared error (RMSE), taking the result of the segment mean as the reference for computing the relative errors.

As it can be observed, SM and STP improve significantly over GM, although it is DM that achieves the best results, particularly in terms of RMSE. This is due to the adaptive nature of this model. In the next subsection, we focus on strictly dynamic aspects of this model.

5.2 Predicting Traffic Jams

The second experiment concerns a specific traffic situation. It focuses on an accident that occurred on 26th of September 2011 in the selected area. The local press reported¹ that day to be unusually affected by traffic jams in the whole city, but specifically in the selected area: a lorry broke down in the Jana Pawla II street and was removed only about 9 p.m., resulting in unusual congestion lasting to late evening hours. The accident coincided with the beginning of a new academic year, during which students return to the city, additionally increasing the traffic.

Concerning the above case we test the performance of the models on two separate days, September 19th and September 26th. The former was chosen to proceed the latter by exactly one week, to be used for reference and comparison with the fairly unusual September 26th. For each of them a learning set spanning a week before the chosen day is extracted and the static models are constructed using these data sets.

Figure 2 (left) shows the RMSE for the static time period model and the dynamic model throughout the two compared days. On September 19th the prediction errors of the static and the dynamic models are similar and

¹ http://poznan.gazeta.pl/poznan/1,36037,10359810,Poznan_sparalizowany_Wina_jednego_tira_.html (“Poznań jammed, one lorry to blame”), in Polish.

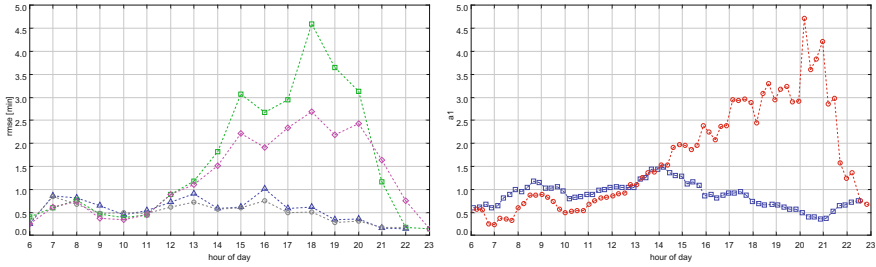


Fig. 2 (Left) Prediction error for models by the time of day, for two different days: \square SM September 26th, \diamond DM September 26th; \triangle SM September 19th, \circ DM September 19th. (Right) Regression coefficient a_1 of the dynamic model by the time of day, for two different days: \square September 19th, \circ September 26th.

fairly low. We suspect that no major incident happened on that day, and the traffic conditions were typical, producing quite accurate predictions of the static model and a slight, but consistent improvement the dynamic model over the static one. On the other hand, the prediction error of both models rises significantly in the afternoon of September 26th. This is most probably caused by the much higher variance in the data, which resulted from the unusual traffic situation. Also in this case it can be seen that as far as the prediction error is concerned, the dynamic model is much better than the static model, as it is able to adapt to the unexpected accident. According to the local press, the broken lorry was removed about 9 p.m., and this can also be seen in the figure, as the prediction error starts to fall down around this hour. The predictions of the dynamic model are a bit worse after 9 p.m., which may be caused by the (decreasing but existent) bias of the previous events that were still present in the model's one-hour time window.

Let us observe the behaviour of the dynamic model by studying the values of the coefficient a_1 , which adapts the static model to the current traffic situation. This coefficient reflects the general traffic congestion. Values greater than 1 suggest the presence of traffic jams, while values lower than or equal to 1 suggest free traffic flow. As it can be seen in Figure 2 (right), the afternoon values of a_1 differ significantly between the two days under consideration. On September 19th, a_1 slightly oscillates around the default value of 1, suggesting that the dynamic traffic conditions match the static, historical pattern. On September 26th, a_1 starts rising in the afternoon, approaching 5 at about 8 p.m., which reflects the extreme and unusual increase of the travel time in the area, lasting well into the evening hours. After 9 p.m., as the incident has been dealt with, the values of a_1 fall rapidly, suggesting the return to normal traffic conditions.

6 Conclusions

The paper presents and discusses drifts in travel time estimates of a floating car data based car navigation system. Starting with dynamic, multi-channel stream data sent in from moving cars, of essentially amorphous form, the system uses static road network information to structuralize these data into a more manageable form and uses them in successful travel time estimation.

Using exclusively real-life data, provided by NaviExpert, it is shown how the travel time estimates change, or drift, in reaction to unusual traffic situations, like unexpected jams caused by sudden road accidents. The presented, combined prediction model, consisting of static and dynamic components, is capable of targeting both the long-term and the short-term prediction horizons and while it is clear that it cannot guarantee a perfect solution to all possible situations, it performs satisfactorily enough to be used in practice.

Acknowledgements. This research is as a part of the project UDA-POIG.01.04.00-30-066/11-00 carried out by NaviExpert Sp. z o. o., co-financed by the European Regional Development Fund under the Operational Programme ‘Innovative Economy’.

References

1. Brosch, P.: A service oriented approach to traffic dependent navigation systems. In: IEEE Congress on Services - Part I, pp. 269–272 (2008)
2. Du, B., Xu, L., Ma, D., Lv, W., Zhu, T.: Missing data compensation model in real-time traffic information service system. In: Fifth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008, vol. 5, pp. 371–378 (2008)
3. de Fabritiis, C., Ragona, R., Valenti, G.: Traffic estimation and prediction based on real time floating car data. In: 11th International IEEE Conference on Intelligent Transportation Systems, ITSC 2008, pp. 197–203 (2008)
4. Gawel, P., Jaskiewicz, A.: Improving short-term travel time prediction for on-line car navigation by linearly transforming historical traffic patterns to fit the current traffic conditions. *Procedia Social and Behavioral Sciences* 20, 638–647 (2011)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter* 11(1), 10–18 (2009)
6. Hastie, T., Tibshirani, R., Friedman, J.H.: *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer (2003)
7. Hiramatsu, A., Nose, K., Tenmoku, K., Morita, T.: Prediction of travel time in urban district based on state equation. *Electronics and Communications in Japan* 92(7), 1–11 (2009)
8. Rice, J., Van Zwet, E.: A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems* 5(3), 200–207 (2004)
9. Van Lint, J., Hoogendoorn, S., Van Zuylen, H.: Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C* 13(5-6), 347–369 (2005)

SONCA: Scalable Semantic Processing of Rapidly Growing Document Stores

Marek Grzegorowski, Przemysław Wiktor Pardel,
Sebastian Stawicki, and Krzysztof Stencel

Abstract. Scientific data constitutes a great asset. However, its volume is far bigger than any human can comprehend. Therefore, automatic analytical, search and indexing solutions are called for. In this paper we present the architecture and the data model of such a system. SONCA (Search based on ONtologies and Compound Analytics) is a platform to implement and exploit intelligent algorithms identifying relations between various types of objects (publications, inventions, scientists and institutions). The results of these algorithms can be used to build semantic search engines but also can be fed into further analytical algorithms in order to find even more associations. We also show experimental evaluation of the performance of SONCA. Its results are promising and we argue that SONCA's architecture is robust.

1 Introduction

Large volumes of scientific data require specific storage and indexing solutions to maximize the effectiveness of searches [6]. Semantic indexing algorithms use domain knowledge to group and rank sets of objects, such as publications, scientists, institutes, and scientific concepts. Their implementation is often based on massively parallel solutions employing NoSQL platforms. Different types of semantic processing operations should be scaled with respect to the growing volumes of scientific information using different database methodologies [2].

Marek Grzegorowski · Przemysław Wiktor Pardel · Sebastian Stawicki · Krzysztof Stencel
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw,
ul. Banacha 2, 02-097 Warsaw, Poland
e-mail: [M.Grzegorowski, Stawicki, Stencel}@mimuw.edu.pl](mailto:{M.Grzegorowski, Stawicki, Stencel}@mimuw.edu.pl)

Przemysław Wiktor Pardel
Institute of Computer Science, University of Rzeszów,
ul. Dekerta 2, 35-030 Rzeszów, Poland
e-mail: ppardel@univ.rzeszow.pl

SONCA (Search based on ONtologies and Compound Analytics) platform is developed at the Faculty of Mathematics, Informatics and Mechanics of the University of Warsaw. It is part of the project Interdisciplinary System for Interactive Scientific and Scientific-Technical Information¹. SONCA is an application based on a hybrid database framework, wherein scientific articles are stored and processed in various forms. SONCA is expected to provide interfaces for intelligent algorithms identifying relations between various types of objects [11]. Presented system utilizes automatic semantic tagging algorithms which extends standard Explicit Semantic Analysis (ESA) method, by enriching it with interactive users feedback processing, to improve quality of the tags that it produces [5]. SONCA is supposed to index documents from multiple, heterogeneous data sources. Such situation causes a lot of data redundancy of created objects(authors, articles, organizations etc.) and poses the problem of data matching and similar entities recognition [14].

The assumptions underlying the construction of the system SONCA involve exposing multiple APIs for users as well as for algorithms and other client systems. According to most recent research results, system is supposed to support two innovative methods of data clustering. One employs Artificial Intelligence algorithms to enrich document vector space model by semantic information to achieve semantic search result clustering [8]. Other proposes custom document grouping based on additional information provided by the user in query [7].

This paper is organized as follows. In Section 2 we present the architecture of the SONCA system. Section 3 describes the schema of the central analytical storage constituent of the system. In Section 4 we summarize experimental evaluation of the performance of SONCA. Using this results we argue that the prototype implementation is robust. Section 5 concludes.

2 Architecture

SONCA’s architecture comprises four layers (cf. Fig. 1). User Interface receives requests in a domain-specific language, rewrites them into the chains of (No)SQL statements executed iteratively against the Semantic Indices and – in some cases – the contents of SoncaDB (Document Database), and prepares answers in a form of relational, graph or XML structures that can be passed to external reporting and visualization modules.

The Semantic Indices are periodically recomputed by Analytic Algorithms based on continuously growing contents of SoncaDB. SoncaDB stores articles (and other pieces of information) acquired from external sources in two formats: XML extracted for each single document using structural OCR, and subsets of tuples corresponding to documents’ properties, parts, single words and relationships to other objects, populated across data tables in a relational database.

¹ See www.synat.pl

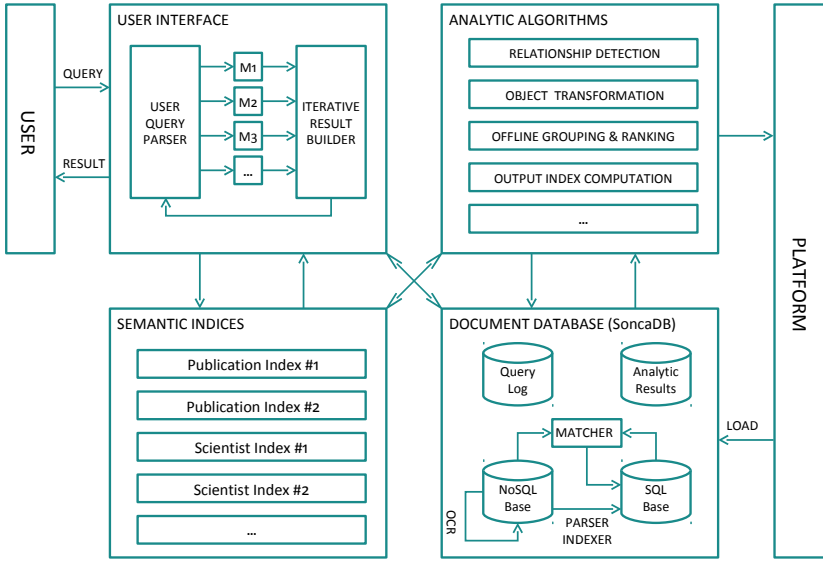


Fig. 1 Major layers of SONCA’s architecture

The roles of Analytic Algorithms and the relational subpart of SoncaDB are two examples of SONCA’s innovation. Additional storage of full information about articles gives developers of Analytic Algorithms a choice between relational, structural and mixed methods of data access, data processing, and storage of the obtained outcomes. However, for millions of documents, we should expect billions of rows. Hence, the underlying RDBMS technologies need to be very carefully adjusted.

The increase of volumes of tuples with each document loaded into SONCA is faster than one might expect. Each article yields entities corresponding to its authors, bibliography items, and areas of science related to thematic classifications or tags. These entities are recorded in generic tables as instances of objects (eg scientist, publication, area) with some properties (eg scientist’s affiliation or article’s publisher) and links to a document from which they were parsed (including information about the location within a document that a given article was cited, a given concept was described and so on). Instances are grouped into classes corresponding to actual objects of interest (for instance: bibliographic items parsed from several documents may turn out to be the same article) using (No)SQL procedures executed over SoncaDB. Analytic Algorithms are then adding their own tuples corresponding, for instance, to heuristically identified semantic relations between objects that improve the quality of search processes.

Owing to the rapid growth in the volume of data we use technologies based on intelligent software rather than massive hardware. We chose Infobright’s analytic

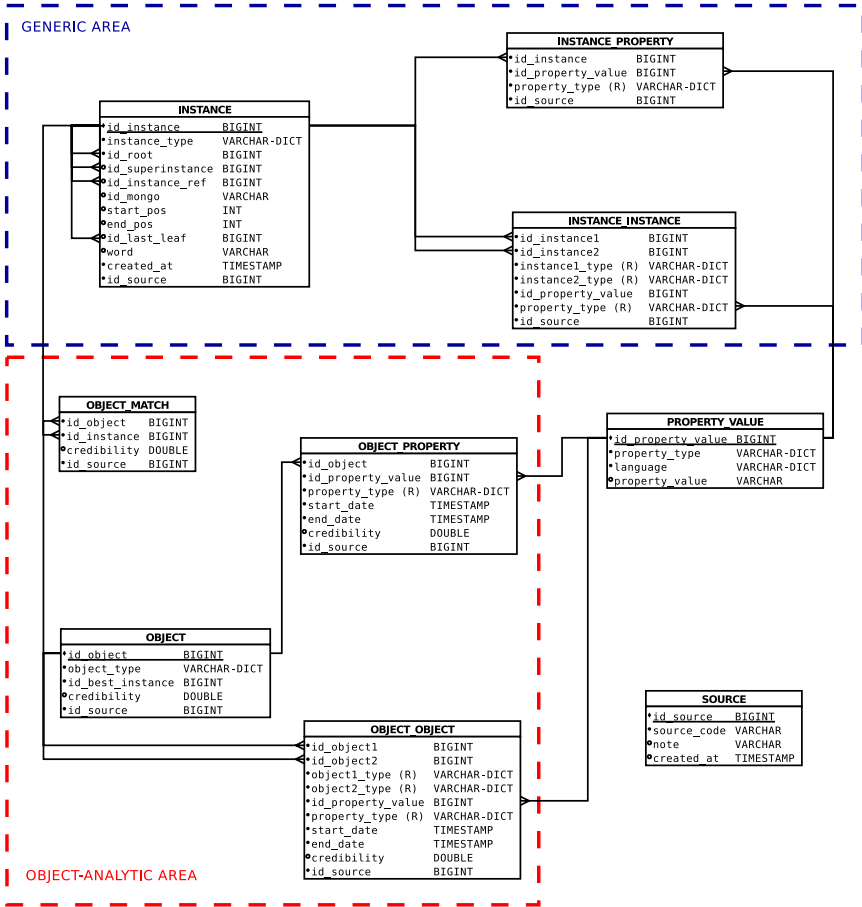


Fig. 2 Relational subpart of SoncaDB - database model

RDBMS engine [12] [13] to handle the relational subpart of SoncaDB because of its performance on machine-generated data originating from sources such as sensor measurements, computer logs or RFID readings. Although SONCA gathers contents created by humans, the way in which they are parsed and extended makes them more similar to machine-generated data sets. We also use carefully selected solutions for other database components such as the structural subpart of SoncaDB (MongoDB is employed here because of its flexibility of enriching objects with new information) and the Semantic Indices (outputs of Analytic Algorithms can be stored in Cassandra, Lucene or PostgreSQL, for example, depending on how they are used by User Interface modules).

3 Data Model of SQL Base

Our research group invested a significant amount of time and effort to gather a corpus of documents containing about 220k full-text journal articles from the PubMed Central Open Access Subset repository and about 1.7M brief entries of full-text articles containing standard information like title, authors, abstract, article classification (according to the ACM Computing Classification System).

The National Center for Biotechnology Information (NCBI) of the National Library of Medicine (NLM) has faced the problem of creating common format for exchanging textual content like articles or books. They prepared Journal Archiving and Interchange Tag Suite which simplifies document structure schema preparation process. From a technical point of view the tag suite is a set of DTD modules which can be combined to create a specific XML tag set. NCBI/NLM has prepared and made available several tag sets for specific purposes. The one on which we focused and which seems to meet our requirements of common format for representing documents in structural subpart of SoncaDB is called Journal Archiving and Interchange Tag Set². SONCA's codename for this format is NXML.

The content of the relational subpart of SoncaDB is populated in the process of parsing documents from structural subpart of SoncaDB. Parsers transform publications stored as NXML format files into tuples corresponding to various parts of documents ranging from the very general like title, authors, abstract, keywords, references, text content to the very specific like single words.

The relational subpart of SoncaDB is designed according to the Entity-Attribute-Value (EAV) pattern. EAV has proven to be effective in machine learning and data mining applications [10]. Fig. 2 presents entity-relationship diagram. Attributes marked by "(R)" are redundant and have been added for performance. Attributes of type VARCHAR-DICT are implemented using efficient Infobright's lookup columns which utilizes integer substitution for values³. The general idea assumes that there are entities in the world and we can observe occurrences of them. This situation poses the following problems: (1) Occurrences of the same entity can be seen as different (eg errors and typos in titles or names). (2) Occurrences of two different entities can be seen as identical or very similar (eg two people have the same name). There are two highlighted areas which are designed to model objects and phenomena observable in the world of documents and address the above problems.

The GENERIC AREA models the document world entities occurrences. It consists of three database entities:

- INSTANCE - represents single occurrence of the documents world entity eg publication, abstract, keyword, author, person, affiliation, organization, word, etc.
- INSTANCE_PROPERTY - stores information specific to a created instance eg a name and a surname of a person instance
- INSTANCE_INSTANCE - stores information about relations between instances eg a person is an author of a publication or an article refers to another article

² See <http://dtd.nlm.nih.gov/archiving>

³ See <http://www.infobright.org>

The OBJECT-ANALYTIC AREA models the document world actual entities. A general view of this area is quite similar to the GENERIC AREA which is expected as entities occurrences may be considered as snapshots of the actual entities. The model area consists of four tables. OBJECT represents single document world entity. OBJECT_PROPERTY contains information specific to an object. OBJECT_OBJECT contains information about relations between objects. OBJECT_MATCH represents information about grouping instances into objects. While the original data acquired as the input to the system is parsed and stored in the GENERIC AREA, the OBJECT-ANALYTIC AREA is to store the results of intelligent algorithms. Such algorithms can be fed with the generic data as well as already collected analytic results. Goals of this algorithms include but are not limited to (1) discovering relationship between objects and instances (also deduplication of acquired base data) and (2) finding new and refining existing properties of objects (eg coalescing and reconciling attributes of matched duplicate instances). Similar concept of object-object relationships can be found in [11].

For completeness, it is necessary to mention about the remaining database entities. PROPERTY_VALUE serves as a container for value of the properties - mentioned above the properties of individual entities (INSTANCE and OBJECT) or properties of relations between entities (INSTANCE_INSTANCE and OBJECT_OBJECT). SOURCE identifies the algorithms runs which led to the creation of the particular database tuples.

To illustrate how we model concepts of documents let us present a simple example. Actual parser input format (NXML format) is out of the scope of the paper. Therefore to avoid unnecessary technical details the following example document is presented in a way that people can easily understand and interpret.

```
Title: Lorem ipsum
Author: Jan Kowalski, University of Warsaw
Year published: 2009
Abstract: Lorem ipsum dolor
Content:
Chapter 1 - Lorem
Lorem ipsum dolor sit amet, consectetur adipisicing elit.
References:
Jan Nowak | Previous lorem | 2005
```

Tables [13] present a parser output important from the point of view of modeling document world entities interactions. Table [1] contains occurrences (instances) of the entities together with their structure. There is a publication (Lorem ipsum) which has subinstances like title, abstract, authors, references, etc. Each of the subinstances have their own subinstances - author has surname and name, reference has its authors, title or year of publication, particular words of its content, etc. Besides instances directly visible in a parsed document there are also indirect occurrences of entities. Encountering an author entity occurrence the algorithm creates also a person entity occurrence or encountering an affiliation it creates an organization instance. Table [2] contains properties of the created instances eg "the instance identified by 'INS1002' has 'surname'/'name' property which actual value is stored as

Table 1 Sample content of the INSTANCES table

id_instance	instance_type	id_super_instance	id_instance_ref	id_root	start_pos	end_pos	id_last_leaf	word
INS0	publication	NULL	NULL	INS0	0	31	INS52	NULL
INS1	title	INS0	NULL	INS0	0	1	INS3	NULL
INS2	word	INS1	NULL	INS0	0	0	INS2	Lorem
INS3	word	INS1	NULL	INS0	1	1	INS3	ipsum
INS4	author	INS0	INS1002	INS0	2	3	INS8	NULL
INS5	name	INS4	NULL	INS0	2	2	INS6	NULL
INS6	word	INS5	NULL	INS0	2	2	INS6	Jan
INS7	surname	INS4	NULL	INS0	3	3	INS8	NULL
INS8	word	INS7	NULL	INS0	3	3	INS8	Kowalski
INS9	afiliation	INS0	INS1003	INS0	4	6	INS12	NULL
INS10	word	INS9	NULL	INS0	4	4	INS10	University
INS11	word	INS9	NULL	INS0	5	5	INS11	of
INS12	word	INS9	NULL	INS0	6	6	INS12	Warsaw
INS13	publish_year	INS0	NULL	INS0	7	7	INS14	NULL
INS14	word	INS13	NULL	INS0	8	8	INS14	2009
INS15	abstract	INS0	NULL	INS0	9	11	INS18	NULL
INS16	word	INS15	NULL	INS0	9	9	INS16	Lorem
INS17	word	INS15	NULL	INS0	10	10	INS17	ipsum
INS18	word	INS15	NULL	INS0	11	11	INS18	dolor
INS19	section	INS0	NULL	INS0	12	21	INS21	NULL
INS20	title	INS19	NULL	INS0	12	12	INS21	NULL
INS21	word	INS20	NULL	INS0	12	12	INS21	Lorem
...
INS31	references	INS0	NULL	INS0	22	26	INS42	NULL
INS32	reference	INS31	INS1004	INS0	22	26	INS42	NULL
INS33	author	INS32	INS1005	INS0	22	23	INS37	NULL
INS34	name	INS33	NULL	INS0	22	22	INS35	NULL
INS35	word	INS34	NULL	INS0	22	22	INS35	Jan
INS36	surname	INS33	NULL	INS0	23	23	INS37	NULL
INS37	word	INS36	NULL	INS0	23	23	INS37	Nowak
INS38	title	INS32	NULL	INS0	24	26	INS41	NULL
INS39	word	INS38	NULL	INS0	24	24	INS39	Previous
INS40	word	INS38	NULL	INS0	25	25	INS40	lorem
INS41	publish_year	INS32	NULL	INS0	26	26	INS42	NULL
INS42	word	INS41	NULL	INS0	26	26	INS42	2005
...
INS1002	person	NULL	NULL	INS1002	NULL	NULL	INS1002	NULL
INS1003	organization	NULL	NULL	INS1003	NULL	NULL	INS1003	NULL
INS1004	publication	NULL	NULL	INS1004	NULL	NULL	INS1004	NULL

Table 2 Sample content of the INSTANCE_PROPERTIES table

id_instance	id_property_value	property_type
INS0	PV0	title
INS0	PV1	publish_year
INS1002	PV2	name
INS1002	PV3	surname
INS1004	PV4	title
INS1004	PV5	publish_year

Table 3 Sample content of the PROPERTY_VALUES table

id_property_value	property_type	property_value
PV0	title	Lorem ipsum
PV1	publish_year	2009
PV2	name	Jan
PV3	surname	Kowalski
PV4	title	Previous lorem
PV5	publish_year	2005

Table 4 Sample content of the INSTANCE_INSTANCE table

id_instance1	id_instance2	instance1_type	instance2_type	id_property_value	property_value
INS1002	INS0	person	publication	NULL	is_author_of
INS1002	INS1004	person	publication	NULL	is_author_of
INS1002	INS1003	person	organization	NULL	is_affiliated_with
INS1004	INS0	publication	publication	NULL	is_referenced_by

a record identified by 'PV2'/'PV3' in property_values table", etc. Table 4 contains relations between instances i.e. "Jan Kowalski is the author of the 'Lorem ipsum' publication", "Jan Kowalski is affiliated with University of Warsaw", etc. Table 3 contains actual values of the example document's parts properties.

4 Performance

The performance and quality tests undertaken so far on over 200K full-content articles resulting in 300M tuples confirm SONCA's scalability [11], which should be investigated not only by means of data volume but also ease of adding new types of objects that may be of interest for specific groups of users.

To demonstrate the SONCA's scalability we prepare tests of the scalability the data loading process and performance tests of SQL queries. The data has been loaded into the test table (according to the scheme SONCA) simultaneously for 1, 2, 4, 6 and 8 indexing processes (tests were performed on the server with 2xIntel (R) Xeon (R) CPU X5650@2.67GHz and 4GB RAM @ 1333MHz). The test results presented in Figure 3 confirm SONCA's scalability.

By increasing the number of indexing processes the loading efficiency was obtained at more than 2,5K documents per hour (maximum size of the loaded XML document is approximately 600KB). Moreover the average loading time of a single publication was respectively 4.8s, 5.0s, 6.3s, 8.5s and 11.2s (Fig. 4).

We can notice that during data processing throughput of the system at first is increasing linearly with the number of simultaneously working processes and slows when server load becomes high. In order to keep the scalability of the system we should also take care of the appropriate increase of hardware. The proposed solution can process data in parallel and give the possibility to scale large amounts of data.

In the next stage of testing we prepared performance tests of SQL queries with the Apache JMeter - designed to load test functional behaviour and measure performance (Results presented in Table 5). The data model performance tests were conducted on test databases of various sizes (from 39M, to over 1762M tuples) and were simulating usage of several users at the same time (1, 2, 3, 4, 5, 6, 10 and 30 users working simultaneously). We decided to query data model in order to reconstruct whole content of the publication but return only the contents of text and

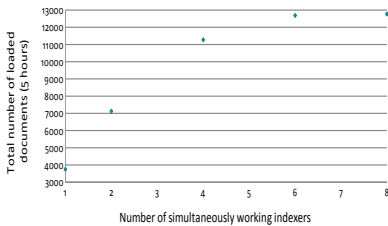


Fig. 3 SONCA tests of the scalability loading process

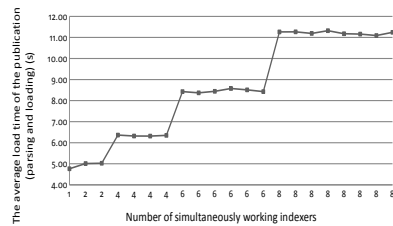


Fig. 4 Average load time of a single publication

Table 5 Average response time of SONCA’s database for document reconstruction query. In columns: number of concurrently working users. In rows: number of tuples in examined database.

no. of tuples (M for millions)	1 user	2 users	3 users	4 users	5 users	6 users	10 users	30 users
39 M	40 ms	43 ms	58 ms	93 ms	110 ms	130 ms	247 ms	841 ms
73 M	36 ms	46 ms	60 ms	76 ms	119 ms	125 ms	250 ms	862 ms
118 M	38 ms	46 ms	61 ms	75 ms	109 ms	133 ms	248 ms	770 ms
132 M	43 ms	47 ms	61 ms	89 ms	115 ms	139 ms	254 ms	839 ms
343 M	45 ms	49 ms	67 ms	78 ms	117 ms	128 ms	244 ms	855 ms
1726 M	46 ms	51 ms	83 ms	98 ms	124 ms	137 ms	251 ms	890 ms

identifiers (to reduce the impact of the data transmission on the test results). To disable database caching, inspected query was run cyclically for 1000 publications identifiers (instance_id) selected from the test database and was performed 1000 times for each user. On the basis of tests we can make a conclusion that the increasing amount of data doesn’t slow down queries response time on the analytical database.

The relational data model employed within SoncaDB enables smooth extension of the set of supported types of objects with no need to create new tables or attributes. It is also prepared to deal on the same basis with objects acquired at different stages of parsing (eg concepts derived from domain ontologies vs. concepts detected as keywords in loaded texts) and with different degrees of information completeness (eg fully available articles vs. articles identified as bibliography items elsewhere). However, as already mentioned, the crucial aspect is the freedom of choice between different data forms and processing strategies while optimizing Analytic Algorithms, reducing execution time of specific tasks from (hundreds of) hours to (tens of) minutes.

5 Conclusion

In this paper we have presented the architecture of our novel analytical platform for intelligent algorithms identifying numerous associations among scientific objects. We have also shown the innovative data model of the main analytical database of the system. Eventually, we summarize the volumes of data already loaded into our system and analyze the performance of SONCA against several analytical queries. The results prove that the system is robust and can scalably serve increasing workload.

SONCA extends typical functionality of scientific search engines by more accurate identification of relevant documents and more advanced synthesis of information. To achieve this, concurrent processing of documents needs to be coupled with ability to produce collections of new objects using specific analytic queries.

The presented framework is to be used together with a number of technologies and methodologies, eg approximate OLAP [4], graphical representation of knowledge [9] and recursive querying [3].

References

1. Adar, E., Teevan, J., Agichtein, E., Maarek, Y. (eds.): Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12. ACM (2012)
2. Agrawal, R., et al.: The claremont report on database research. *Commun. ACM* 52(6), 56–65 (2009)
3. Burzańska, M., Stencel, K., Suchomska, P., Szumowska, A., Wiśniewski, P.: Recursive Queries Using Object Relational Mapping. In: Kim, T.-H., Lee, Y.-H., Kang, B.-H., Słezak, D. (eds.) FGIT 2010. LNCS, vol. 6485, pp. 42–50. Springer, Heidelberg (2010)
4. Cuzzocrea, A., Serafino, P.: LCS-hist: taming massive high-dimensional data cube compression. In: Kersten, M.L., Novikov, B., Teubner, J., Polutin, V., Manegold, S. (eds.) EBDT. ACM International Conference Proceeding Series, vol. 360, pp. 768–779. ACM (2009)
5. Janusz, A., Świeboda, W., Krasuski, A., Nguyen, H.S.: Interactive Document Indexing Method Based on Explicit Semantic Analysis. In: Yan, J., et al. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 156–165. Springer, Heidelberg (2012)
6. Kersten, M.L., Manegold, S.: Revolutionary database technology for data intensive research. *ERCIM News* (89) (2012)
7. Meina, M.: Query-context search result clustering basing on graphs. In: Szczuka, M., Czaja, L., Skowron, A., Kacprzak, M. (eds.) CS&P, Puttusk, Poland, pp. 346–352. Białystok University of Technology (2011) Electronic edition
8. Nguyen, S.H., Świeboda, W., Jaśkiewicz, G.: Extended Document Representation for Search Result Clustering. In: Bembienik, R., Skonieczny, Ł., Rybiński, H., Niezgodka, M. (eds.) Intelligent Tools for Building a Scient. Info. Plat. SCI, vol. 390, pp. 77–95. Springer, Heidelberg (2012)
9. Poelmans, J., Ignatov, D., Kuznetsov, S., Dedene, G., Elzinga, P., Viaene, S.: Formal concept analysis in knowledge processing: A survey on applications. *Inf. Sci.* (2012)
10. Sarawagi, S., Thomas, S., Agrawal, R.: Integrating association rule mining with relational database systems: Alternatives and implications. *Data Min. Knowl. Discov.* 4(2/3), 89–125 (2000)
11. Słezak, D., Janusz, A., Świeboda, W., Nguyen, H.S., Bazan, J.G., Skowron, A.: Semantic Analytics of PubMed Content. In: Holzinger, A., Simoncic, K.-M. (eds.) USAB 2011. LNCS, vol. 7058, pp. 63–74. Springer, Heidelberg (2011)
12. Słezak, D., Synak, P., Borkowski, J., Wróblewski, J., Toppin, G.: A rough-columnar RDBMS engine – a case study of correlated subqueries. *IEEE Data Eng. Bull.* 35(1), 34–39 (2012)
13. Słezak, D., Wróblewski, J., Eastwood, V., Synak, P.: Brighthouse: an analytic data warehouse for ad-hoc queries. *PVLDB* 1(2), 1337–1345 (2008)
14. Szczuka, M., Betliński, P., Herba, K.: Named Entity Matching in Publication Databases. In: Yan, J., et al. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 172–179. Springer, Heidelberg (2012)

Collective Classification Techniques: An Experimental Study

Tomasz Kajdanowicz, Przemyslaw Kazienko, and Marcin Janczak

Abstract. Collective classification is the area in machine learning, in which unknown nodes in the network are classified based on the classes assigned to the known nodes and the network structure only. Three collective classification algorithms were described and examined in the paper: Iterative Classification (ICA), Gibbs Sampling (GS) and Loopy Belief Propagation (LBP). Experiments on various networks revealed that greater number of output classes decreases classification accuracy, GS provides better results than ICA and LBP outperforms others for dense structures while it is worse for sparse networks.

1 Introduction

One of the most important direction in research on networks is node classification. Classification of nodes in the network may be performed either by means of known profiles of these nodes (regular concept of classification) or based on information derived from the interconnection of nodes in the network - collective classification. An example of such problem is web page categorization based on categories of pages connected to it. It is very likely that a given web page is related to sport, if it is linked by many other web pages about sport. In general, there have been proposed several types of collective classification approaches. Three of them: Iterative Classification Algorithm (ICA), Gibbs Sampling Algorithm (GSA) and Loopy Belief Propagation (LBP) are examined and presented in the paper in order to assess their predictive accuracy on several datasets from distinct domains.

The paper is organized as follows: the concise presentation of related work in the field of collective classification is presented in Section 2. In Section 3 there

Tomasz Kajdanowicz · Przemyslaw Kazienko · Marcin Janczak
Wroclaw University of Technology, Department of Computer Science and Management
e-mail: tomasz.kajdanowicz@pwr.wroc.pl, przemyslaw.kazienko@pwr.wroc.pl,
marcinjanczak88@wp.pl

are described three examined algorithms. The experimental results and comparison of the methods together with evaluation of the algorithms' accuracy for different contributions of the known labels in the entire network are gathered in Section 4 and concluded in Section 5.

2 Related Work

There exist a variety of methods for collective classification. However, it can be distinguished two distinct types of them: local and global. The former methods use a collection of local conditional classifiers successively applied to the unknown nodes whereas the latter are defined as optimization of one global objective function.

Additionally classification of nodes in network can be solved using two distinct approaches: within-network and across-network inference. Within-network classification [2], for which training nodes are connected directly to other nodes, whose labels are to be classified, stays in contrast to across-network classification [7], where models learnt from one network are applied to another similar network.

There are related several problems with collective classification that have been currently addressed by researchers. One of them is the problem of what features should be used to maximize the classification accuracy. In approaches which use local classifiers the relational domain needs to be transformed to standard notation by application of proper aggregation operator. It has been reported that precise solution strongly depends on the application domain [11]. The previous research showed that new attribute values derived from the graph structure of the network, such as the betweenness centrality, may be beneficial to the accuracy of the classification task [3]. It was also confirmed by other research discussed in [5].

Another interesting problem in collective classification based on iterative algorithms is the ordering strategy that determines, in which order to visit the nodes iteratively to re-label them. The order of visiting the nodes influences the values of input features that are derived from the structure. A variety of sophisticated or very simple algorithms can be used for this purpose. Random ordering that is one of the simplest ordering strategies used with iterative classification algorithms can be quite robust [6].

Two of the most popular local collective classification methods are: Iterative Classification Algorithm (ICA) and Gibbs Sampling Algorithm (GSA), introduced by Geman & Geman in the image processing context [4]. Both of them belong to so called approximate local inference algorithms basing on local conditional classifiers [12]. Another technique is a Loopy Belief Propagation (LBP) [10] that is the global approximate inference method used for collective classification. As in the literature it was not found the comparison of predictive accuracy of mentioned methods across datasets from distinct domains we examine these algorithms using eight distinct datasets and present the comparison in the paper.

3 Collective Classification Techniques

3.1 Iterative Classification

The basic idea behind ICA is quite simple but reasonable. Considering a node $v_i \in V^{UK}$, where V^{UK} is a set of nodes with unknown label, $V^{UK} \subset V$, we aim to discover its label l_i . Having known labels of v_i 's neighbourhood ICA utilizes a local classifier Φ that takes the attribute values of nodes with known labels (V^K) and returns the best label value for v_i from the class label set L . If the knowledge of the neighbouring labels is partial the classification process needs to be repeated iteratively. In each iteration labelling of each node v_i is done using current best estimates of local classifier Φ and continues until the label assignments are stabilized. A local classifier might be any function that is able to accomplish the classification task. It can range from a decision tree to an SVM in its place.

Algorithm 1 depicts the ICA algorithm as a pseudo-code where the local classifier is trained using the initially labelled nodes V^K only. It can be observed that the attributes utilized in classification depend on current label assignment (lines 8 and 9 in Algorithm 1). Thus there need to be performed the repetition of classification phase until labels stabilize or maximal number of iteration is reached.

Algorithm 1. Iterative Classification Algorithm (ICA), the idea based on [12]

```

1: for each node  $v_i \in V^{UK}$  do
2:   compute  $x_i$ , i.e.  $v_i$ 's attributes using observed nodes  $V^K$ 
3: end for
4: train classifier  $\Phi$  by  $\Theta$  optimization using attributes of  $V^K$  nodes
5: repeat
6:   generate ordering  $O$  over nodes in  $V^{UK}$ 
7:   for each node  $v_i \in O$  do
8:     compute  $x_i$ , i.e.  $v_i$ 's attributes using current assignments
9:      $l_i \leftarrow \Phi(x_i, \Theta)$ 
10:  end for
11: until label stabilization or maximal number of iterations

```

3.2 Gibbs Sampling

Gibbs Sampling Algorithm (GSA) is used in the collective classification assuming that a local classifier is trained using initial network data and is accessible in iterative label settlement process.

As presented in Algorithm 2 GSA contains four main steps: bootstrapping, burn-in, samples collection and final computation of labels. Bootstrapping phase results with trained classifier Φ obtained from optimization of its Θ parameters. In burn-in

step, performed s times, all nodes are classified according to generated ordering O . It is common case that random ordering is being chosen as O . Attributes of nodes with unknown labels V^U in each iteration are updated. Note that the recalculation of features is made on the updated graph, i.e. on $G^{(n)}$. In the third phase of the algorithm it is performed a collection of classification results for each node. Labels for each $v_i \in O$ are sampled as well as counted, i.e. we obtain information about how many times label l was sampled for node v_i . After collecting a number of samples and fulfilling a stop criterion, label l_i appearing mostly in sampling are assigned separately to each of the nodes $v_i \in V^U$. Maximum label count is a usual criterion of label assignment.

Algorithm 2. Gibbs Sampling Algorithm (GSA), the idea based on [12]

```

1: //bootstrapping
2: for each node  $v_i \in V^{UK}$  do
3:   compute  $x_i$ , i.e.  $v_i$ 's attributes using  $V^K$ 
4: end for
5: train classifier  $\Phi$  by  $\Theta$  optimization using  $V^K$ 
6: for each node  $v_i \in V^U$  do
7:    $l_i \leftarrow \Phi(x_i, \Theta)$ 
8: end for
9: //burn-in
10: for  $n = 1$  to  $s$  do
11:   generate ordering  $O$  over nodes in  $V^U$ 
12:   for each node  $v_i \in O$  do
13:     compute  $x_i$  attributes using  $G^{(n)}$ 
14:      $l_i \leftarrow \Phi(x_i, \Theta)$ 
15:   end for
16: end for
17: //initialize sample counts
18: for each node  $v_i \in V^{UK}$  do
19:   for label  $l \in L$  do
20:      $c[i, l] = 0$ 
21:   end for
22: end for
23: //collect samples
24: repeat
25:   generate ordering  $O$  over nodes in  $V^U$ 
26:   for each node  $v_i \in O$  do
27:     compute  $x_i$ 's attributes using the updated  $G$ 
28:      $l_i \leftarrow \Phi(x_i, \Theta)$ 
29:      $c[i, l] \leftarrow c[i, l] + 1$ 
30:   end for
31: until stop condition
32: //compute final labels
33: for each node  $v_i \in O$  do
34:    $l_i \leftarrow \operatorname{argmax}_{l \in L} c[i, l]$ 
35: end for

```

3.3 Loopy Belief Propagation

Loopy Belief Propagation (LBP) is an alternative approach to perform collective classification in comparison to ICA and GSA. The main difference is that it defines a global objective function to be optimized, instead of performing local classifiers optimization.

Intuitively, LBP is an iterative message-passing algorithm. The messages are transferred between all connected nodes v_i and v_j , $v_i, v_j \in V$, $(v_i, v_j) \in E$, and might be interpreted as belief of what v_j label should be based on v_i label.

The global objective function that is optimized in the LBP is derived from the idea of pairwise Markov Random Field (pairwise MRF) [13]. In order to calculate the message to be propagated the calculation presented in Equation 1 is performed.

$$m_{i \rightarrow j}(l_j) = \alpha \sum_{l_i \in L} \Psi_{ij}(l_i, l_j) \phi(l_i) \prod_{v_k \in V^{UK} \setminus v_j} m_{k \rightarrow i}(l_i) \quad (1)$$

where $m_{i \rightarrow j}(l_j)$ denotes a message to be sent from v_i to v_j , α is the normalization constant that ensures each message sum to 1, Ψ and ϕ denotes the clique potentials. For further explanation see [12].

The calculation of believe can be concisely expressed as in Equation 2:

$$b_i(l_i) = \alpha \phi(l_i) \prod_{v_j \in V^{UK}} m_{j \rightarrow i}(l_i) \quad (2)$$

The LBP algorithm consist of two main phases: message passing that is repeated until the messages are stabilized and believe computation, see Algorithm 3.

Algorithm 3. Loopy Belief Propagation (LBP), the idea based on [12]

```

1: for each edge  $(v_i, v_j) \in E, v_i, v_j \in V^{UK}$  do
2:   for each class label  $l_i \in L$  do
3:      $m_{i \rightarrow j}(l) \leftarrow 1$ 
4:   end for
5: end for
6: //perform message passing
7: repeat
8:   for each edge  $(v_i, v_j) \in E, v_i, v_j \in V^{UK}$  do
9:     for each class label  $l_i \in L$  do
10:       $m_{i \rightarrow j}(l_j) \leftarrow \alpha \sum_{l_i \in L} \Psi_{ij}(l_i, l_j) \phi(l_i) \prod_{v_k \in V^{UK} \setminus v_j} m_{k \rightarrow i}(l_i)$ 
11:    end for
12:  end for
13: until stop condition
14: //compute beliefs
15: for all  $v_i \in V^{UK}$  do
16:   for all  $l_i \in L$  do
17:     $b_i(l_i) \leftarrow \alpha \phi(l_i) \prod_{v_j \in V^{UK}} m_{j \rightarrow i}(l_i)$ 
18:   end for
19: end for

```

4 Experimental Study

4.1 Experimental Scenarios

In order to evaluate the predictive accuracy of ICA, GSA and LBP algorithms a new experimental environment has been developed in Java language. All datasets were stored using Pajek data format. As ICA and GSA required classification algorithm C4.5 decision tree has been used as a base classifier. The experiments were carried out on original dataset with primary prepared splits between nodes with known and unknown labels. Each dataset was split into known and unknown node sets in nine distinct proportions (from 10% to 90% unknown labels). The split was accomplished by node sampling using uniform distribution. In order to assess distinct classification approaches standard measure of classification accuracy was recorded.

4.2 Datasets

The experiments were carried out on eight datasets. The AMD_NETWORK graph presents seminary attendance at conference. The dataset was a result of a project that took place during The Last HOPE Conference held in July 18-20, 2008, New York City, USA. At this conference RFID (Radio Frequency Identification) devices were distributed among participants and allowed to uniquely identify them and track in which sessions they attended. The data set is build from information about descriptions of interests of participants, their interactions via instant messages, as well as their location over the course of the conference. Location tracking allowed to extract a list of attendances for each conference talk. In general, the most interesting for experiment information included in dataset are: information about conference participants, conference talks and presence on talks. The genealogy dataset CS_PHD is the network that contains the ties between Ph.D. students and their advisers in theoretical computer science where arcs points from an advisers to a students [9]. The dataset NET_SCIENCE contains a co-authorship network of scientists working

Table 1 Basic properties of datasets utilized in experiments

Dataset	Nodes	Edges	Classes	Avg. node degree
AMD_NETWORK	332	69092	16	208.108
ARTIFICIAL	413	415	6	1.004
CRN	327	324	4	0.990
CS_PHD	1451	924	16	0.636
NET_SCIENCE	1588	2742	26	1.726
PAIRS_FSG	4931	61449	3	12.461
PAIRS_FSG_SMALL	1972	12213	3	6.193
YEAST	2361	2353	13	0.996

on network theory and experiment [8]. It was extracted from the bibliographies of two review articles on networks. The biological dataset YEAST consists of protein-protein interaction network [1]. The PAIRS dataset is a dictionary from The University of South Florida word association, rhyme, and word fragment norms. This graph presents correlation between nouns, verbs and adjectives. It was used original data as well as its small version. Additionally, collective classification approaches were examined on two artificially generated graphs: CRN and ARTIFICIAL. These datasets were created according to simple sampling procedure constructing edges between nodes in accordance to the frequency of given class label in whole dataset. Namely if the the node is of a frequent class it has small degree and if the class is rare it has high degree. For CRN and ARTIFICIAL dataset we used 4 and 6 classes respectively, with highly skewed distribution. The profiles of the datasets were shortly depicted in Tab. 1.

Table 2 Accuracy of ICA, GSA and LBP collective classification algorithms obtained for eight datasets. Results are presented for consequent percentages of unknown labels in the graph.

Dataset	Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
AMD_NETWORK	ICA	0.105	0.094	0.094	0.092	0.086	0.083	0.098	0.083	0.087
	GS	0.116	0.097	0.098	0.106	0.101	0.097	0.101	0.093	0.099
	LBP	0.152	0.129	0.122	0.122	0.108	0.108	0.104	0.104	0.105
ARTIFICIAL	ICA	0.331	0.385	0.346	0.374	0.383	0.374	0.351	0.364	0.335
	GS	0.395	0.389	0.403	0.405	0.402	0.399	0.384	0.386	0.344
	LBP	0.376	0.371	0.381	0.435	0.462	0.453	0.458	0.444	0.410
CRN	ICA	0.479	0.491	0.512	0.533	0.556	0.531	0.546	0.533	0.539
	GS	0.513	0.509	0.540	0.596	0.583	0.587	0.600	0.615	0.617
	LBP	0.506	0.432	0.481	0.534	0.542	0.483	0.483	0.547	0.534
CS_PHD	ICA	0.256	0.263	0.249	0.300	0.316	0.267	0.289	0.310	0.317
	GS	0.273	0.306	0.298	0.315	0.320	0.318	0.317	0.317	0.333
	LBP	0.123	0.100	0.094	0.086	0.086	0.081	0.087	0.071	0.066
NET_SCIENCE	ICA	0.097	0.088	0.092	0.091	0.093	0.095	0.084	0.090	0.087
	GS	0.103	0.094	0.098	0.095	0.097	0.098	0.099	0.093	0.093
	LBP	0.076	0.076	0.072	0.074	0.072	0.071	0.068	0.065	0.063
PAIRS_FSG	ICA	0.712	0.715	0.725	0.737	0.680	0.620	0.690	0.645	0.651
	GS	0.767	0.768	0.746	0.738	0.709	0.696	0.692	0.693	0.690
	LBP	0.736	0.751	0.758	0.761	0.752	0.742	0.726	0.710	0.680
PAIRS_FSG_SMALL	ICA	0.578	0.580	0.542	0.531	0.526	0.454	0.368	0.318	0.284
	GS	0.640	0.620	0.617	0.582	0.539	0.486	0.407	0.339	0.333
	LBP	0.641	0.630	0.641	0.646	0.632	0.624	0.604	0.574	0.537
YEAST	ICA	0.312	0.325	0.310	0.292	0.247	0.211	0.249	0.237	0.207
	GS	0.328	0.341	0.321	0.300	0.274	0.253	0.251	0.252	0.245
	LBP	0.236	0.224	0.179	0.181	0.193	0.158	0.129	0.135	0.155

4.3 Results

The accuracy values for various contribution of known nodes (from 10% to 90%), for all three classification algorithms (ICA, GS, and LBP) were presented in Tab. 2 and Fig. 1.

As we can see the average accuracy is at different level for various datasets. For the NET_SCIENCE dataset, it exceeds 10% only once, whereas for PAIRS_FSG, it is regularly above 70%. Overall, better results can be achieved if the problem is simpler, i.e. the greater the number of classes the worse results. it means that the quality of collective classification, like other, regular classification methods, strongly depends on the problem and sometimes it is hardly to obtain very good results.

In almost all cases Iterative Classification (ICA) outperforms Gibbs Sampling (GS). Loopy Belief Propagation (LBP) works worse than the other algorithms for the sparse networks, i.e. with the small average degree value about 1 (CS_PHD, YEAST, CRN, NET_SCIENCE) and boosts its results for dense networks - average degree above 6 as for AMD_NETWORK, both PAIRS datasets. These differences

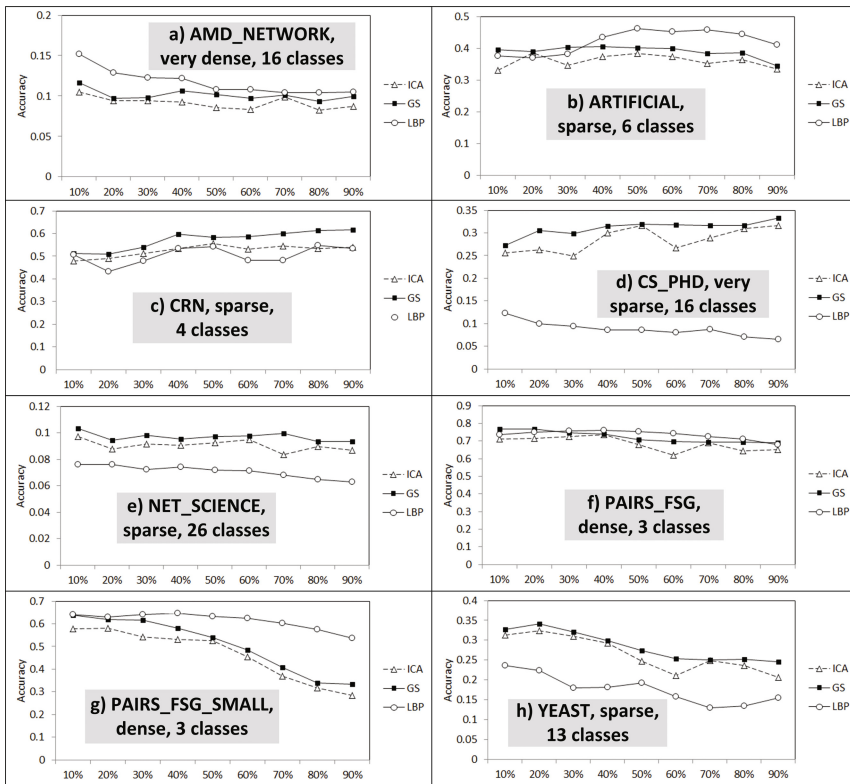


Fig. 1 Accuracy of collective classification (ICA, GSA and LBP) in relation to distinct contributions of unknown nodes in the network for 8 datasets

were smaller for artificial datasets (CRN and ARTIFICIAL) than for real ones. The difference in accuracy for smaller contribution of unknown nodes (e.g. 10%) and for most nodes unlabelled (90%) is not significant.

5 Conclusions and Future Works

The main goal of the paper was to investigate various algorithms for classification of nodes in the network - collective classification algorithms. In particular three of them were presented and examined: Iterative Classification (ICA), Gibbs Sampling (GS) and Loopy Belief Propagation (LBP).

The general conclusion derived from the experiments carried out on 8 datasets is as follows: ICA works a bit better than GS, LBP outperforms others for dense networks and it is worse for sparse structures, better results can be obtained in case of smaller number of classes.

The future works will focus on deeper analysis of results for different network topologies as well as on efficiency of algorithms.

Acknowledgements. This work was partially supported by The Polish National Center of Science the research project 2011-2012, 2011-2014 and Fellowship co-financed by The European Union within The European Social Fund.

References

1. Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Lu, H., Zhang, J., Sun, S., Ling, L., Zhang, N., Li, G., Chen, R.: Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research* 31(9), 2443–2450 (2003)
2. Desrosiers, C., Karypis, G.: Within-Network Classification Using Local Structure Similarity. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) *ECML PKDD 2009*. LNCS, vol. 5781, pp. 260–275. Springer, Heidelberg (2009)
3. Gallagher, B., Eliassi-Rad, T.: Leveraging label-independent features for classification in sparsely labeled networks: An empirical study. In: *Proceedings of Second ACM SIGKDD Workshop on Social Network Mining and Analysis, SNA-KDD 2008* (2008)
4. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721–741 (1984)
5. Kazienko, P., Kajdanowicz, T.: Label-dependent node classification in the network. *Neurocomputing* 75(1), 199–209 (2012)
6. Knobbe, A.J., de Haas, M., Siebes, A.: Propositionalisation and Aggregates. In: Siebes, A., De Raedt, L. (eds.) *PKDD 2001*. LNCS (LNAI), vol. 2168, pp. 277–288. Springer, Heidelberg (2001)
7. Lu, Q., Getoor, L.: Link-based classification. In: *Proceedings of 20th International Conference on Machine Learning, ICML, San Francisco*, pp. 496–503 (2003)
8. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. *Physical Review E* 74, 036104 (2006)

9. Nooy, W., Mrvar, A., Batagelj, V.: Exploratory Social Network Analysis with Pajek, ch. 11. Cambridge University Press (2004)
10. Pearl, J.: Probabilistic reasoning in intelligent systems. Morgan Kaufmann (1988)
11. Perlich, C., Provost, F.: Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning* 62(1-2), 65–105 (2006)
12. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. *Artificial Intelligence Magazine* 29(3), 93–106 (2008)
13. Taskar, B., Abbeel, P., Koller, D.: Discriminative probabilistic models for relational data. In: *Proceedings of 18th Conference in Uncertainty in Artificial Intelligence*. Morgan Kaufmann, San Francisco (2002)

Granular Knowledge Discovery Framework*

A Case Study of Incident Data Reporting System

Adam Krasuski, Dominik Ślęzak, Karol Kreński, and Stanisław Łazowy

Abstract. A platform for fire & rescue incident data reporting system (IDRS) is presented as an example how the domain knowledge driven granule formation can assist in knowledge discovery and decision support. The current modeling, monitoring and reporting systems rarely take advantage of semantic background of the analyzed phenomena. We discuss how to build and tune practically meaningful models of processes by means of granules approximating their states and instances. We show how the layers of model creation should interact with lower-level layers of data preparation and transformation. We illustrate the proposed methodology by several IDRS related use cases. We also discuss the complexity of available data sources that can be utilized to make the proposed approach more useful.

Keywords: Knowledge Discovery, Domain Knowledge, Granular Modeling, Layered Architectures, Fire Services, Text Data, Heterogeneous Data Sources.

1 Introduction

The national fire & rescue services are typically equipped with the incident data reporting systems (IDRS), which gather the information about the conducted actions. Implementations of IDRS remain usually at the level of simple reporting, with no attempt to model domain knowledge related to the risk and logistics of rescue

Adam Krasuski · Karol Kreński · Stanisław Łazowy
The Main School of Fire Service, ul. Słowackiego 52/54, 01-629 Warsaw, Poland
e-mail: {krasuski, krenski, lazowy}@inf.sgsp.edu.pl

Dominik Ślęzak
Institute of Mathematics, University of Warsaw, ul. Banacha 2, 02-097 Warsaw, Poland,
Infobright Inc., ul. Krzywickiego 34, 02-078 Warsaw, Poland
e-mail: slezak@infobright.com

* Partly supported by the Polish National Science Centre grant 2011/01/B/ST6/03867.

actions, and with no possibility to organize and analyze data in a truly meaningful way. Semantically driven data organization or, in other words, granulation is important in order to reason about events that are both well-supported in available data sets and easy to understand by the users [1]. It is important to operate with bigger clusters or granules of objects in order to assign them with statistics that reflect the model's types and dynamics. The usage of domain knowledge in order to build granules with both semantically and statistically meaningful descriptions is the key to create models of complex real-world phenomena, in a process that one may call as the *granular knowledge discovery*.

In this paper, we propose how to extend the existing Polish version of IDRS, named EWID¹, in order to let it derive useful knowledge from large amounts of operational reports gathered in a semi-structured form. We show that the usage of domain knowledge can help during data cleaning and transformation, which means, e.g., detection and proper handling of outliers, or defining meaningful features annotating both unstructured and structured data parts. We claim that formation of semantically useful features needs to be conducted in parallel to identification of granules that represent the states of modeled processes according to the domain experts. We also claim that even those attributes that occur in the structured data parts may reflect efficient data acquisition rather than semantics of models required by the users. One may refer to this difference as to the *semantic gap* that occurs in many areas of decision support and knowledge discovery [2].

The paper is organized as follows. Section 2 provides more detailed motivation for our approach. Section 3 outlines four layers of our platform: the raw data layer, the quality data layer, the granular layer, and the models layer. Although we discuss them specifically for IDRS, they may be reused also for other granular knowledge discovery applications. Section 4 includes several use cases of models related to potential requirements of the IDRS users, such as commanders' efficiency model, blockage management model, monitoring of rescue capability and detection of abnormal reports, as well as the decision support for the commander. In all those cases, an emphasis is put on the importance of operating with sufficiently meaningful granules while constructing the corresponding models. Section 5 provides an insight into future research directions and concludes the paper.

2 Motivation for the Proposed Platform

Most of our knowledge discovery process is consistent with the KDD stages, i.e., selection, preprocessing, transformation, data mining, interpretation / evaluation [3], paying a special attention to the interaction with the experts at each of stages.

In our research, we adopted the experts' reasoning paths in order to resolve particular problems. We realized that the experts do not utilize the raw data by a simple selection of the important features. They rather use abstract data representations,

¹ <http://www.ewid.pl/>

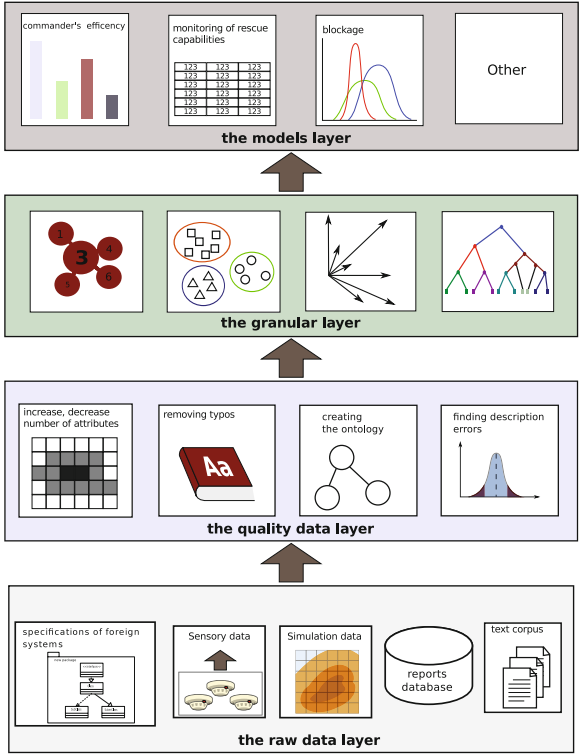


Fig. 1 The proposed layers of the IDRS platform

such as *within the group of similar incidents*, *density of the housing* or *heavy equipment* and there are no such attributes in the database. Sometimes, the experts refer to the entities which actually exist in the database, such as *time*, but they usually use their own quantifications such as *in the afternoon* or *early morning*. The conclusion is that our models derived from domain knowledge need to operate on abstract data in order to match experts' representations. This leads towards creation of the *granular layer*, which aims at holding the abstract concepts.

The abstract concepts gathered within the granular layer need to be derived from the *raw data layer*, which contains raw data stream or layer of a poor quality. There are not just typographic errors in the raw data. – There is no semantic information about the content, which hampers construction of granules. Therefore, the *quality data layer* was introduced in between. This is how the platform gets a natural multilayered shape, with the *models layer* at the top level. Such a platform, illustrated by Figure 1, will enable to define models driven from domain knowledge, and exploring multiple levels of granularity in problem solving.

3 Incidents Data Reporting System

3.1 Models Layer

This layer is a placeholder for the models built to answer the needs of the end users. The models layer is an algorithmic framework which deals with knowledge discovery, decision support, data consistency, problems prediction, etc. and will work on the basic or abstract structures provided by the granular layer. The methods from the fields of machine learning, data clustering, regression analysis, peculiarity mining, etc. [4, 5, 6, 7] will be used to achieve the desired functionalities.

In our previous research, we considered the following models: estimating commanders' efficiency, blockage management, monitoring of the rescue capabilities and abnormal reports detection. The first two of them have been already implemented. The description of the above models follows in Section 4.

3.2 Granular Layer

The purpose of the granular layer is to serve the input for the models layer. This layer provides some benefits over just simple data layer. The granular layer is where indirect, often abstract data are collected in form of information granules.

Real-world problems are usually defined by the users in abstract form, as opposed to anything readily available from the database. A question about any single attribute (EWID database contains over 560 attributes) is less likely than a question which requires some extra operation on attributes. This is how granules are created.

As it is impossible to foresee all the future demands for the abstract concepts, this layer will be progressively expanded with new granules. A new granule is built by analyzing the essence of the demanded information. This process is analogous to the way of perceiving and building the complex concepts by the experts. The actual operations on the data usually include the composition of the attributes and their quantization. Moreover, the granular layer may serve as a domain definition generator, as our previous research revealed that some concepts which may seem basic to the domain experts do not have clear definitions (e.g. *large fires*).

We are not focused on inventing new granules out of nothing, but we are sensitive to hints coming with the demands from the models layer. If the demand comes for the granule *large fires*, then we may consider building the granules *small fires* and *medium fires* as well. We gradually extend the layer and to properly manage it we need to provide some meta data, likely in the form of an ontology.

Some mechanisms of validation of the granular layer are needed too [8]. The granules may not always come out best. Validation of the granule quality may trigger an automatic or semi-automatic improvement process. Another aspect of self-organization is searching and perhaps filling the granule gaps – once the granules *large fires* and *small fires* are created, it may be revealed that there are data which belong somewhere in between, hence the hint for *medium fires* is provided.

3.3 *Quality Data Layer*

This layer contains good quality data. The main concern with the raw data is that there is no sufficient validation mechanism for the user input. There are roughly 4% of typographic errors, inconsistent punctuation impeding identification of sentences' boundaries, and there is no control over synonyms.

The typographic errors are addressed by collecting and using a set of dictionaries (some general like *aspell* and some domain oriented). The performance of the corrections depends on assigning weights to the dictionaries and on analysing the neighbourhoods of the misspellings (n-gram analysis).

Surely, the quality data layer requires also more advanced techniques. There are around 560 attributes in EWID database and we need to find whether there are any excessive or missing attributes. The indication of the excess would be that irrespectively of the values of some attributes the analysis would return mostly the same results. Some missing attributes have already been revealed (e.g. weather data) and more are expected when constructing models of real-world problems.

Some focus should be also given to the data semantics. The important concepts used in the corpus need to be indicated and organized in the knowledge base. The process should be oriented on finding the lists of synonyms and the general strategy is to have the descriptions standardized as much as possible. We choose ontology as the tool for the semantic enrichment of the data and the actual ontology is currently being developed by human experts. Since this work is quite laborious, it is narrowed to a subset of the whole corpus, namely the fires of the buildings. It allows for naming relations between concepts and organizing the data – the main reason for creating this ontology is to annotate each single action report with a set of practically meaningful concepts.

The quality data layer needs to be prepared for *sensors*, which are the sources of real-time data streams [9]. These streams of data must be processed and become semantically shaped. We propose three main sensors: a) the devices installed in the buildings (e.g. fire detectors), b) the natural language reports from the actual action field provided by the officer in charge, c) the computer simulations. Each of such sensors provides raw data, like the voltage levels from the fire detectors, which need to be transformed to some more meaningful entities. Another useful application could be mapping of the sensory data to the physical conditions found in the given locations in the building. The independent data from the multiple sensors need to be integrated to achieve a better picture of the overall situation [10].

3.4 *Raw Data Layer*

This layer gathers data coming from various sources. It is designed with the openness for importing additional data, but at the current stage the key source is the database of the Polish IDRS named EWID. Following is the list of the data sets which may be also considered: a) text corpus containing around 3.000 articles from

science and fire protection domain journals, b) fire protection forums, c) Polish-English fireman's e-dictionary, d) American database of fireman's life threat incidents, e) rescue actions methods described online in English, f) specification of the abroad IDRS systems, g) fire & rescue wiki created by us and our students (approximately 2.400 records).

The collected text corpus can be used for finding concepts, building ontology or enrich short descriptions in the natural language section. Creating ontologies to model knowledge for a given domain is now a mainstream approach to achieve better logical organization of data and better search and modeling abilities [11].

We also consider other types of data outlined in the previous subsection. For instance, the already-mentioned fire detectors are installed in most of the buildings of public infrastructure. Currently, such detectors operate in a binary mode. However, by the use of special adapters it is possible to obtain a continuous flow of data providing the information about the amount and the type of the smoke or the temperature for the given location in the building.

Another type of sensors considered in our further work are virtual sensors, which can be modeled by using *computational fluid dynamics* approach [12]. Such an approach allows to model the fire parameters and dynamics inside the buildings. Currently used models are still not excellent, but we hope that with the support of the domain knowledge we will be able to generate valuable results.

During the rescue actions the commander communicates with his/her crew and the control room staff. Such a communication – as mentioned in the previous subsection as well – may be considered as the stream of words. This implies that the commander and other persons involved in the fire & rescue action may be considered sensors too. For the sake of our research, we suggest introducing the concept of *semantic sensor* to represent the sources of such data. We shall attempt to utilize such data in order to reflect the conditions and dynamics of the fire as perceived by the commander and other participants of the rescue action.

4 Examples of Specific Models

4.1 Commanders' Efficiency Model

In this section we provide a couple of use cases, as well as how we benefit from the proposed platform and what needs to be extended in the future. We start with the model evaluating the officers in charge and their actions based on information collected in IDRS. The evaluation criterion is the distribution of durations of actions. Precisely, the measure for comparing the commanders is the *operating time* of the conducted action. The operating time is measured from the arrival of the commander at the scene until the end of the action.

Different commanders may conduct actions under different circumstances. Also, a single commander may conduct multiple actions under different circumstances. Without injecting domain knowledge how to discriminate between different

action circumstances, the model of commanders' comparison and evaluation would be quite misleading. This is why, in our approach, the granules of similar cases were first created. Then, within each of such granules, sub-granules grouping the incidents conducted by each single commander were identified. For each of such sub-granules the distribution of operating time was calculated and the adequate model was fitted – the log-normal was chosen as a universal model for all the distributions. After the regression, the distributions of operating times of different commanders could be finally compared. By studying the resulting curves we can describe both commanders' efficiency (the mean) and experience (the standard deviation).

The proposed model proves that IDRS allows for more advanced and interesting statistics of this sort, as opposed to just basic statistics currently made. Description of the model can be found in [13]. It may be further enhanced by redefining the quality-of-action-measure and increasing the semantic and statistical quality of granules. With regards to the semantic aspect of quality, it is important to involve domain experts into verification of granule descriptions. With regards to the statistical aspect of quality, some questions concerning a choice of applied measures and an equal treatment of single objects may arise.

The considered framework may be also extended towards predicting the commanders' efficiency. In this case, the statistics of above-discussed sub-granules may play a role of decision values, while descriptions of the above granules of similar actions and available information about particular commanders may be transformed into attributes in a training data sample. Analogous studies were undertaken, e.g., in medical data analysis [4]. In our case, an additional challenge is that at least some part of attributes corresponding to granule descriptions needs to be derived from domain knowledge about the available unstructured data sets.

4.2 *Blockage Management*

This model is focused on handling the blockages in the fire stations. Blockage refers to the situation when all fire units are out and a new incident occurs. Our major contribution relates here to an effective solution for supporting the decision making for the dynamic deployment of the fire & rescue units across the fire stations.

The approach is as follows: the large collection of the IDRS data is first split into granules of the similar accidents. For each of the granules the distribution of operating times is calculated. The aim is to obtain the probability distribution of time needed to handle each distinct emergency situation. Once the probability distribution is found, it becomes possible to estimate in real time when the gone units are expected to return to the fire station. This can be further extended to monitor for blockage probability when the fire station gets short on the units reserves.

The results prove that data granulation significantly improves the accuracy of prediction. Description of this model can be found in [14].

4.3 *Detection of Abnormal Reports*

The fire service headquarters of Poland issued a problem of spotting the invalid reports, i.e. reports with invalid incident size or type, or invalid number of firemen. It is usually caused by the typos or oversight of people entering the incident reports. The existing validation mechanism is not sufficient as it is implemented as a manually defined set of rules resulting mostly from formal regulations (e.g.: *small-sized fire is the fire to which up to 4 fire fighting jets have been applied*).

Unfortunately, the majority of the data fields in reports can take any value and are not dependent on other fields or regulations, therefore they cannot be easily checked against a defined list. Currently no such reports are detected as invalid which causes the data pollution and can affect further data processing stages.

Spotting of the outliers can be alternatively performed by checking how unusual these new records are against the given data set. This may be done by using the techniques such as the already-mentioned peculiarity oriented mining [7], where peculiarities have two fundamental properties: they concern small subset of data and affected data are very different from other data in a data set.

4.4 *Online Decision Support*

It is the largest but also the most interesting challenge of our project. Within the scope of this task we attempt to create the decision support system that would be helpful for commanders during fire & rescue actions in the real time.

At almost every stage of the fire & rescue actions the commander is surrounded by the sea of information. The sources of potential information may be knowledge base of previous accidents, description of the construction of the building involved, sensory data from the systems and installations within the building and many other depending on the type of the building. However, the commander is not able to collect and process that amount of information. In most cases the amount of information is too large and too complex for human processing. Therefore, during the actions the commander uses his/her experience and intuition in order to properly carry out his/her activities [15]. Moreover, he does not have a chance to share online his/her experience and intuition in order to ensure the accuracy of his/her assumptions.

The main idea which arises as a result of these considerations is to create the computer system which collects all the available data, converts, processes and sends them to the commander in suitable structure and granulation. Unfortunately, until now the successful combination of such resources of information in order to support the commanders was not reported in the literature.

There is a number of difficulties in facing the problem of developing such a system. One of them is the problem of collecting all of the information and processing them online. Another arising problem is how we should communicate with the commander in order to not disturb him by asking for the details of the action.

Our primary idea is to model some processes offline. As an input for this modeling would be, for example, knowledge base of previous accidents, the fire simulations and others. The output of this process would be the fire evolving scenarios. We can produce many scenarios offline. The choice of the proper scenario during the rescue actions will depend on the sensory data.

In our approach, the sensory data could be continuous data sent by e.g. fire detectors as well as a stream of words in the correspondence of the commander with the control room staff and other rescuers.

Treating communication between the commander and other rescuers as semantic sensors is a step forward in improving the communication between humans and system. We assume that we will be able to extract important concepts from the stream of words and send them to the system in order to adjust the selected scenario. Surely, this means communication, which fits rather higher layers of Figure 11.

Such an approach defines the need for the processing of the large data streams online. Processing the continuously incoming data implies new computational requirements concerning limited amount of the memory and the short processing time. Moreover, our further data processing will be done in non-stationary environments, where the underlying data distribution may change over time. All these conditions and also the aspects of online decision making, creates a very complex decision-making environment. This imposes a need to seek for new solutions in the areas such as *knowledge discovery from data streams*, *concept drift* and many others.

5 Conclusions

In this paper, we proposed a multilayered platform, which enables development of the models derived from the knowledge of domain experts. By introducing the granular layer, which feeds the abstract data to the models layer, the platform attempts to follow the humans' way of problem solving.

The granular layer allows for building more robust models for the real-world phenomena, as well as for more advanced and interesting representations of the modeled realms. The presented use cases illustrate that granular computing is indeed a powerful perspective that can be used to model problems.

One of important aspects is how to verify the quality of the elements of the granular layer. Such evaluation can be considered at: 1) the semantic level, by letting the experts validate the abstract granules against their real-world counterparts, 2) the data level, by evaluating granules as data clusters using statistical and machine learning measures, as well as 3) the model level, by means of the practical quality of the models based on the analyzed granules.

Another important challenge is how to integrate all sources of data that can be useful for constructing granular models in such complex applications as those related to the IDRS systems. In our opinion, a special emphasis should be put in the future on the *semantic sensors*, which have the potential for providing the context for processing other types of data.

References

1. Bargiela, A., Pedrycz, W.: *Granular Computing: An Introduction*, vol. 717. Springer (2003)
2. Moss, L.T., Atre, S.: *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-support Applications*. Addison-Wesley (2003)
3. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. *AI Magazine* 17(3), 37 (1996)
4. Bazan, J.G., Skowron, A., Ślęzak, D., Wróblewski, J.: Searching for the Complex Decision Reducts: The Case Study of the Survival Analysis. In: *International Symposium on Methodologies in Intelligent Systems*, Maebashi, Japan, October 28-31, pp. 160–168 (2003)
5. Hand, D.J.: *Statistics: A Very Short Introduction*. Oxford University Press (2008)
6. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. *ACM Computing Surveys* 31(3), 264–323 (1999)
7. Yang, J., Zhong, N., Yao, Y., Wang, J.: Local Peculiarity Factor and Its Application in Outlier Detection. In: *Knowledge Discovery in Databases*, pp. 776–784 (2008)
8. Szczuka, M., Ślęzak, D.: Representation and Evaluation of Granular Systems. In: Watada, J., Watanabe, T., Phillips-Wren, G., Howlett, R.J., Jain, L.C. (eds.) *Intelligent Decision Technologies*. SIST, vol. 15, pp. 287–296. Springer, Heidelberg (2012)
9. Gama, J.: *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC (2010)
10. Babitski, G., Bergweiler, S., Hoffmann, J., Schön, D., Stasch, C., Walkowski, A.C.: Ontology-Based Integration of Sensor Web Services in Disaster Management. In: Janowicz, K., Raubal, M., Levashkin, S. (eds.) *GeoS 2009*. LNCS, vol. 5892, pp. 103–121. Springer, Heidelberg (2009)
11. Kreński, K., Krasuski, A., Łazowy, S.: Data Mining and Shallow Text Analysis for the Data of State Fire Service. In: *Concurrency, Specification and Programming - XXth International Workshop, CS&P 2011*, Pułtusk, Poland, September 28-30, pp. 313–321 (2012)
12. Patankar, S.: *Numerical Heat Transfer and Fluid Flow*. Series in Computational Methods in Mechanics and Thermal Sciences, vol. 67 (1980)
13. Krasuski, A., Kreński, K., Łazowy, S.: A Method for Estimating the Efficiency of Commanding in the State Fire Service of Poland. *Fire Technology*, 1–11 (2011)
14. Krasuski, A., Kreński, K., Wasilewski, P., Łazowy, S.: Granular Approach in Knowledge Discovery: Real Time Blockage Management in Fire Service. In: Li, T., Nguyen, H.S., Wang, G., Gryzma-Busse, J., Janicki, R., Hassanien, A.E., Yu, H. (eds.) *RSKT 2012*. LNCS (LNAI), vol. 7414, pp. 416–421. Springer, Heidelberg (2012)
15. Gadowski, A., Bologna, S., Costanzo, G., Perini, A., Schaerf, M.: Towards Intelligent Decision Support Systems for Emergency Managers: The IDA Approach. *International Journal of Risk Assessment and Management* 2(3), 224–242 (2001)

Diversity in Ensembles for One-Class Classification

Bartosz Krawczyk

Abstract. One-class classification, known also as learning in the absence of counterexamples, is one of the most challenging problems in the contemporary machine learning. The scope of the paper focuses on creating a one-class multiple classifier systems with diverse classifiers in the pool. An approach is proposed in which an ensemble of one-class classifiers, instead of a single one, is used for the target class recognition. The paper introduces diversity measures dedicated to the selection of such specific classifiers for the committee. Therefore the influence of heterogeneity assurance on the overall classification performance is examined. Experimental investigations prove that diversity measures for one-class classifiers are a promising research direction. Additionally the paper investigates the lack of benchmark datasets for one-class problems and proposes an unified approach for training and testing one-class classifiers on publicly available multi-class datasets.

Keywords: one-class classification, machine learning, multiple classifier system, diversity, classifier selection, one-class benchmarks.

1 Introduction

Usually for a given problem we may have a pool of several classifiers at our disposal. Canonical machine learning methods concentrate on selecting the single best classifier from the pool and delegating him to the problem solving task. This approach seems very reasonable and is rooted in a normal human behavior - when having a problem we tend to search for the most competent expert in a given area, not paying attention to lesser renown specialists.

Bartosz Krawczyk

Department of Systems and Computer Networks, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

e-mail: bartosz.krawczyk@pwr.wroc.pl

Yet when referencing to only a single classifier we discard the fact that other models from the pool may also offer a valuable contribution to the considered problem. That is why a combined approach was proposed, utilizing decisions of more than one classifier. Such methods, known as Multiple Classifier Systems (MCS) are considered as one of the most promising research directions in current field of machine learning and pattern recognition [10].

The concept behind the idea of MCSs lies on exploiting the strengths and local competencies of a given pool of classifiers, while at the same time reducing their individual weaknesses. Properly conducted combination of several predictors may give better results than any single one of them. As designing a MCS is not a trivial task, there are several things that one must bear in mind during such a procedure:

- Classifier selection - one should carefully select classifiers that will be included in the ensemble. Wrong choice may even significantly deteriorate the overall system performance. On the other hand using all available classifiers is usually not the best idea. This step will be discussed in more details further in the paper.
- Topology proposal - one needs to choose the nature of interconnections between classifiers in the ensemble. Most widely used is the parallel approach, in which each of classifiers works independently. It has a good methodological background and is used in this work.
- Fuser design - having the outputs of the individual classifiers it is mandatory to choose a procedure of fusing their answers in order to generate the global MCS decision. Fuser should be designed in such a way that can exploit the strengths of the selected classifiers and combine them optimally.

This paper concentrates on the first raised issue, i.e. classifier selection. Let's notice that combining similar classifiers could not contribute much to the system being constructed, apart from increasing the computational complexity. That is why it is important to assure the heterogeneity of the ensemble. One could use several different methods to ensure the diversity of the classifier pool [1]. At the same time there is a need for measures that allow to grade the diversity of a given committee [14]. One may also take into the account the exploitation cost of available classifiers [12].

The previously mentioned issues are well researched for canonical multi-class classifiers. Yet the problem of building MCSs on the basis of one-class still awaits for proper attention. There are several papers dealing with combination of one-class classifiers [19], but most of them are oriented on the practical application [6], not on theoretical advances. Best to author's knowledge there are no works that deal with measuring the diversity and selecting classifiers to the ensemble for such a specific machine learning problem.

This work introduces diversity measures dedicated to one-class problems. Additionally it points out the lack of dedicated benchmark datasets for one-class classification, which poses a difficulty for establishing standards in

one-class MCSs research. Therefore a simple unified scheme for transforming multi-class benchmarks into their one-class equivalents is proposed.

This paper is a continuation of author's work on one-class classifier ensembles [13].

2 One-Class Classification

OCC seeks to distinguish one specific class from the more broad set of classes (e.g., selecting carrot from vegetables, recognizing obstructive nephropathy from various kinds of kidney disorders or identifying medicine-related pictures from an extensive image database). The target class is considered as a positive one, while all other are considered as negative ones. OCC is known as learning in the absence of counterexamples, as primary object of OCC is to train a classifier using only patterns drawn from the target class distribution. Its main goal is to detect anomaly or a state other than the one for the target class [18]. It is assumed that only information of the target class is available. Various terms have been used in the literature to refer to one-class learning approaches. The term single-class classification originates from [11], but also outlier detection [9] or novelty detection [3] are used to name this field of study.

2.1 Combining One-Class Classifiers

There are few works which researches how to combine several one-class predictors. In this paper two fusion methods, proposed in [17] will be used.

One-class boundary methods (such as One-class Support Vector Machine) are based on computing the distance between the object x and target class ω_T . To apply fusion methods we require the probability (or classification support) of object x for a given class. Therefore to conduct the fusion a heuristic mapping must be made. This paper uses a following solution:

$$\hat{P}(x|\omega_T) = \frac{1}{c_1} \exp(-d(x|\omega_T)/c_2), \quad (1)$$

which models a Gaussian distribution around the classifier, where $d(x|\omega_T)$ is a squared Euclidean distance, c_1 is the normalization constant and c_2 is the scale parameter. Parameters c_1 and c_2 should be fitted to the target class distribution.

After performing such a mapping one may use proposed fusion functions. This paper applies two of them, with the assumption that the pool consists of R one-class classifiers:

1. **Mean vote**, which combines binary output labels of one-class classifiers. It is expressed by:

$$y_{mv}(x) = \frac{1}{R} \sum_k I(P_k(x|\omega_T) \geq \theta_k), \quad (2)$$

where $I(\cdot)$ is the *indicator function* and θ_k is a classification threshold. When a threshold equal to 0.5 is applied this rule transforms into a majority vote for binary problems.

2. **Product combination of the estimated probabilities**, which is expressed by:

$$y_{pc}(x) = \frac{\prod_k P_k(x|\omega_T)}{\prod_k P_k(x|\omega_T) + \prod_k \theta_k}. \quad (3)$$

This fusion method assumes that the outlier object distribution is independent of x and thus uniform in the area around the target concept.

3 Benchmarks for One-Class Classification

Most of papers dealing with the theory of machine learning and pattern recognition base their experiments on publicly available datasets, especially from the most popular UCI Repository [5]. This guarantees the repeatability of experiments and allows researchers to compare their results and to judge if their proposal improved something in comparison to other methods. Reproducible experiments are one of the most important issues in the theoretical advancements of machine learning. Yet so far there have been no benchmarks for one-class datasets and only solution on how to assess the performance of new methods for this field was to treat one-class problems as a decomposition of multi-class benchmarks, as in author's previous work [13]. This paper introduces a simple, unified and effective scheme for transformation of multi-class sets into canonical one-class problems.

Let's assume that a dataset consist of objects drawn form class vector $\mathcal{M} = \{1, 2, \dots, M\}$. Class $m \in \mathcal{M}$ is chosen to become target class ω_T . All other objects from $\widehat{\mathcal{M}} = \{1, 2, \dots, M\} \setminus \{m\}$ become outliers objects with labels ω_O . The pool of R individual classifiers $\Psi^{(1)}, \Psi^{(2)}, \dots, \Psi^{(R)}$ is then trained with normal procedure (such as cross-validation) on the objects from class ω_T while objects from ω_O are used for the testing phase. One should notice that from a single M -class problem this procedure may derive M separate one-class datasets. Exemplary transformation, training and testing procedure for two-dimensional five class problem is presented in Fig. 1.

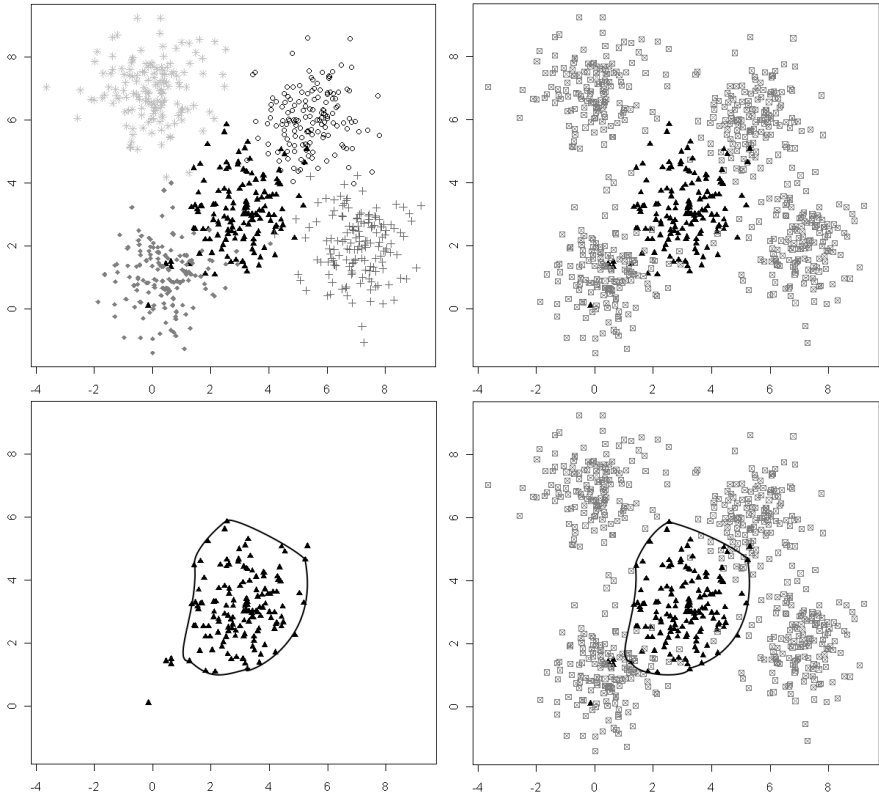


Fig. 1 (*top left*) Original five-class dataset; (*top right*) Selected class transformed into the target and remaining classes into outliers; (*bottom left*) Classifier training on the target class; (*bottom right*) Classifier testing with a test fold from the target class and outliers.

To assess the performance of tested one-class method on such prepared benchmarks two well-known measures, sensitivity and specificity, are proposed, modified with the respect to the nature of considered classification task:

$$Sensitivity_{oc} = \frac{true\ targets}{true\ targets + false\ outliers}, Specificity_{oc} = \frac{true\ outliers}{true\ outliers + false\ targets}. \quad (4)$$

4 Diversity Measures for One-Class Ensembles

In this section three diversity measures dedicated to one-class classifier ensembles are proposed. All of them are based on the concept of *non-pairwise* measure, i.e. such a measure that ranks the diversity of a given pool of

classifiers, not the diversity between a pair of classifiers. In author's opinion such types of measures are more useful for one-class classifier ensembles, as a classifier that did not received a good score in *pairwise* test may still contribute to the overall ensemble. Therefore *non - pairwise* measures return more global outlook on the ensemble performance. All presented diversity measures may take values from $[0,1]$. 0 corresponds to identical ensemble and 1 corresponds to the highest possible diversity.

4.1 One-Class Entropy Measure

Let's assume that the highest ensemble diversity for a given object $x_j \in X$ is displayed by $\lfloor R/2 \rfloor$ of the ensemble votes with the same value (ω_T or ω_O) and remaining $R - \lfloor R/2 \rfloor$ with the other value. If all votes returned identical response the ensemble cannot be considered as a diverse one. Let us denote by $r(x_j)$ the number of one-class classifiers that correctly recognize the object x_j . Assuming there are N objects in the training set, one may use entropy [14] to measure the diversity using the presented concept:

$$E_{oc}(\Pi^r) = \frac{1}{N} \sum_{j=1}^N \frac{1}{(R - \lfloor R/2 \rfloor)} \min\{r(x_j), R - r(x_j)\}. \quad (5)$$

where Π^r is the considered pool of classifiers.

4.2 One-Class Shanon Measure

Fuzzy measures of diversity are a very promising but still largely unexplored area of research. This paper proposes a one-class modification of Fuzzy Shanon diversity measure [7]. Let's assume that there are R classifiers in the pool, out of which S classifiers can correctly classify a given training object $x_j \in X$ to ω_T .

Therefore one may propose a fuzzy membership function $\mu_{x_j} = (S/R)$ for a given object, where $0 \leq (S/R) \leq 1$. Then obtained membership function is given to Shanon function to measure its fuzziness, which acts as a diversity measure:

$$S_{oc}(\Pi^r) = \{(x_j, \mu_{x_j}) | x_j \in X\} \rightarrow \sum_{j=1}^N \{-\mu_{x_j} \text{Ln}(\mu_{x_j}) - (1 - \mu_{x_j}) \text{Ln}(1 - \mu_{x_j})\}. \quad (6)$$

4.3 One-Class Energy Measure

Energy approach is an effective measure of fuzziness, successfully implemented in many practical applications such as ECG analysis [4]. Up to author’s knowledge this measure was not used so far for measuring the diversity of ensembles. Let’s assume identical membership function as in Sec. 4.2. Additionally a threshold $\lambda \in [0, 1]$ is introduced. It’s role is to filter insignificant degrees of membership, that may otherwise contribute to decreasing the stability of the proposed measure. The energy measure is described as follows:

$$EN_{oc}(\Pi^r) = \{(x_j, \mu_{x_j}) | x_j \in X\} \rightarrow \sum_{j=1}^N f_\lambda(\mu_{x_j}), \quad (7)$$

where

$$f_\lambda(x) = \begin{cases} f(x) & \text{if } x \in (\lambda, 1) \\ 0 & \text{if otherwise} \end{cases}, \quad (8)$$

and $f(x) : [0, 1] \rightarrow R_+$ is an increasing function in interval $[0, 1]$ for $f(0) = 0$.

5 Experimental Investigations

5.1 Set-Up

The aim of experiments was to asses the performance of proposed diversity measures and compare their results with a canonical approach for one-class classification.

For the experiments five benchmarks from [5] were used. Their details are presented in Tab. 1. They were subject to the transformation presented in Sec. 3. As all used datasets were binary, therefore out of five benchmarks ten new one-class problems were produced.

As a base classifier a One-class Support Vector Machine (OCSVM) [16] with a polynomial kernel was selected. The pool of classifiers were constructed using a random subspace method [8] and consisted of five models trained on

Table 1 Details of datasets used in the experimental investigation. Numbers in parentheses indicates the number of objects in the minor class.

No.	Name	Objects	Features	Classes
1	Breast-cancer	286 (85)	9	2
2	Diabetes	768 (268)	8	2
3	Heart-statlog	270 (120)	13	2
4	Ionosphere	351(124)	34	2
5	Voting records	435 (168)	16	2

random number of features (subspaces sizes were allowed to differ). Classifiers were selected with the usage of proposed diversity measures through an exhaustive search over all possible combinations.

For energy measure the threshold λ was set to 0.1 and a hyperbolic tangent was selected as the $f(x)$ function.

Table 2 Accuracy of the proposed methods over five benchmarks datasets, transformed into ten one-class problems

No.	Target	Reference methods		Diversity measures				
		Single	Fuser	All	E	S	EN	
1	+1	Sens _{oc}	87.23	Mean	84.32	89.11	87.75	88.61
				Prod	85.36	89.11	88.21	87.89
	Spec _{oc}	79.94	Mean	72.11	78.41	77.15	80.24	
			Prod	75.20	79.94	78.41	78.41	
	-1	Sens _{oc}	91.23	Mean	88.50	93.10	91.80	92.30
				Prod	86.23	91.23	90.02	90.87
	Spec _{oc}	69.50	Mean	64.92	69.50	71.05	71.68	
			Prod	66.21	71.45	70.20	72.28	
2	+1	Sens _{oc}	84.43	Mean	79.05	87.81	85.92	87.02
				Prod	80.11	88.01	84.11	86.55
	Spec _{oc}	82.11	Mean	81.43	82.34	81.90	83.00	
			Prod	82.11	83.00	82.68	83.21	
	-1	Sens _{oc}	92.34	Mean	90.23	93.11	90.85	92.00
				Prod	89.14	92.80	90.10	89.85
	Spec _{oc}	57.10	Mean	51.12	56.55	56.00	58.15	
			Prod	54.20	58.20	57.10	59.35	
3	+1	Sens _{oc}	87.50	Mean	84.00	89.15	90.20	90.20
				Prod	84.00	89.15	90.20	90.20
	Spec _{oc}	84.00	Mean	82.50	86.24	85.31	86.24	
			Prod	82.50	86.24	85.31	86.24	
	-1	Sens _{oc}	90.25	Mean	89.11	88.29	90.00	90.00
				Prod	89.11	88.29	90.00	90.00
	Spec _{oc}	77.98	Mean	71.43	77.02	75.23	75.23	
			Prod	71.43	77.02	75.23	75.23	
4	+1	Sens _{oc}	75.83	Mean	71.87	79.09	77.60	79.53
				Prod	72.45	80.00	78.30	80.00
	Spec _{oc}	88.05	Mean	87.50	88.50	87.20	88.05	
			Prod	87.50	89.05	87.68	87.95	
	-1	Sens _{oc}	94.20	Mean	92.43	95.00	93.95	94.20
				Prod	91.91	95.00	92.80	93.55
	Spec _{oc}	68.45	Mean	60.60	71.20	70.75	70.75	
			Prod	63.45	70.62	70.16	70.16	
5	+1	Sens _{oc}	88.25	Mean	85.11	87.20	88.00	86.95
				Prod	84.25	86.55	87.21	86.25
	Spec _{oc}	79.43	Mean	74.69	80.56	80.05	81.76	
			Prod	75.73	81.87	80.21	82.34	
	-1	Sens _{oc}	95.03	Mean	92.25	95.12	95.03	94.80
				Prod	92.25	95.12	95.03	94.80
	Spec _{oc}	70.23	Mean	65.54	74.28	72.52	75.15	
			Prod	71.76	75.34	74.06	74.94	

As reference methods a single OCSVM and a pool consisting of all classifiers were used. The combined 5x2 cv F test [2] was carried out to assess the statistical significance of the obtained results.

All experiments were carried out in the R language [15].

5.2 Results and Discussion

Results of the experiments are presented in Tab. 2. Labels +1 and -1 denotes which class was chosen as ω_T after the transformation procedure. +1 stands for a minority class (presented in parentheses in Tab. 1) and -1 stands for a majority class. Bolded results shows cases in which a proposed diversity measure was significantly better than results obtained by a single classifier.

In most cases the diversity measures were not worse than a single classifier, even often outperforming it. This is caused by the selection of mutually complementary classifiers to the pool. Therefore using more than one classifier lead to a better decision boundary, when a single model generated too generic solution. In several cases an ensemble with pool consisting of classifiers selected by diversity measure was not as good as a single classifier. This is caused by a fact that diversity measure itself is not the sole determinant of the accuracy. Probably in such cases classifiers with high diversity but low quality were chosen to an ensemble.

Comparing between proposed diversity measures one may see that best results are returned mostly by entropy and energy measures. This fact is not surprising, as entropy measure have proven itself for multi-class problems and energy measure is well-established in the fuzzy systems research area.

6 Final Remarks

The paper addressed two important issues connected with the design of one-class ensembles: a lack of diversity measures for selecting classifiers to the committee and a lack of benchmark datasets dedicated to one-class problems.

Firstly a simple and unified transformation procedure was proposed for extracting one-class benchmarks from multi-class datasets. With this came the proposition of two measures to assess the performance of tested methods. Secondly three diversity measures, dedicated to one-class ensembles were introduced. Two of them were modifications of measures previously used for multi-class problems, while the third was a novel one, based on a fuzzy measure used in ECG processing. Results of computer experiments had proven the high quality of the proposed measures and confirmed that developing compound one-class classifiers is a promising research direction.

In future author would like to concentrate on proposing new measures for classifier selection that take under consideration at the same time ensemble diversity and accuracy, apply proposed methods for imbalanced data classification, examine the influence of fusion methods on one-class ensembles performance and investigate the problem of combining fuzzy one-class classifiers.

References

1. Aksela, M., Laaksonen, J.: Using diversity of errors for selecting members of a committee classifier. *Pattern Recognition* 39(4), 608–623 (2006)
2. Alpaydin, E.: Combined 5 x 2 cv f test for comparing supervised classification learning algorithms. *Neural Computation* 11(8), 1885–1892 (1999)
3. Bishop, C.M.: Novelty detection and neural network validation. *IEE Proceedings: Vision, Image and Signal Processing* 141(4), 217–222 (1994)
4. Czogala, E., Leski, J.: Application of entropy and energy measures of fuzziness to processing of ecg signal. *Fuzzy Sets and Systems* 97(1), 9–18 (1998)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml>
6. Giacinto, G., Perdisci, R., Del Rio, M., Roli, F.: Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Inf. Fusion* 9, 69–82 (2008)
7. Golestani, A., Azimi, J., Analoui, M., Kangavari, M.: A new efficient fuzzy diversity measure in classifier fusion. In: *Proceedings of the IADIS Conference*, pp. 722–726 (2007)
8. Ho, T.K.: The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8), 832–844 (1998)
9. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. *Artificial Intelligence Review* 22(2), 85–126 (2004)
10. Mao, J., Jain, A.K., Duin, P.W.: Statistical pattern recognition: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22(1), 4–37 (2000)
11. Koch, M.W., Moya, M.M., Hostetler, L.D., Fogler, R.J.: Cueing, feature discovery, and one-class learning for synthetic aperture radar automatic target recognition. *Neural Networks* 8(7-8), 1081–1102 (1995)
12. Krawczyk, B., Woźniak, M.: Designing Cost-Sensitive Ensemble – Genetic Approach. In: Choraś, R.S. (ed.) *Image Processing and Communications Challenges 3*. AISC, vol. 102, pp. 227–234. Springer, Heidelberg (2011)
13. Krawczyk, B., Woźniak, M.: Combining Diverse One-Class Classifiers. In: Corchado, E., Snašel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012*. LNCS (LNAI), vol. 7209, pp. 590–601. Springer, Heidelberg (2012)
14. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning* 51(2) (2003)
15. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2008) ISBN 3-900051-07-0
16. Schölkopf, B., Smola, A.J.: *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press (2002)

17. Tax, D.M.J., Duin, R.P.W.: Combining One-Class Classifiers. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 299–308. Springer, Heidelberg (2001)
18. Tax, D.M.J., Duin, R.P.W.: Characterizing one-class datasets. In: Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa, pp. 21–26 (2005)
19. Wilk, T., Wozniak, M.: Soft computing methods applied to combination of one-class classifiers. *Neurocomput.* 75, 185–193 (2012)

Evaluation of Stream Data by Formal Concept Analysis

Martin Radvanský, Vladimír Sklenář, and Václav Snášel

Abstract. Following article presents practical usage of the Formal Concept Analysis (FCA) for the evaluation of stream data recorded during a technological process. The main aim of this paper is to show possibilities of using FCA to detect anomalies in the data. Our attitude is based on the fact that although during the production process a large amount of input data is obtained, the size of conceptual lattice is relatively small, and therefore, it is possible to work with it in real-time. The conceptual lattice represents a model of production process, and this model is based on historical production data. The input data stream contains measurements on the production line and it is applied on the model of the production process. The result of this activity is to identify anomalies in the incoming data and their relationship with faulty products, including disclosure of possible causes of errors and also to obtain a histogram of quality for manufactured products.

Keywords: stream data, quality of production, formal concept analysis, data mining.

1 Introduction

Evaluation of quality of production is an important task in every company. We can check quality during the manufacturing process (online prediction) and during the offline assessment and determination of the cause of defects of products. The quality evaluation is also important while finding the links between physical (chemical ...)

Martin Radvanský · Václav Snášel

VSB Technical University Ostrava, Ostrava, Czech Republic

e-mail: [martin.radvansky.st, vaclav.snasel}@vsb.cz](mailto:{martin.radvansky.st, vaclav.snasel}@vsb.cz)

Vladimír Sklenář

Pike Automation s.r.o., Praha, Czech Republic

e-mail: vsklenar@pikeautomation.cz

variables that can be monitored during the production. In the case of technological processes there is often a large amount of data collected repeatedly with a very short time between measurements (several times per second).

Data representing the production process have the character of stream data. In normal (fault-free) production, it is possible to describe the stream with relatively few patterns. In most cases the patterns are due to the values required by the norms and standards, which must meet the technological process. The actual control of the individual values does not reveal any problematic situation. For example, the wrong combination of values that individually comply may result in poor production. Anomalies in the stream usually signal an issue. The instant recognition of these issues is very important because it enables us to react and prevent damage. It is possible to obtain patterns of the standard production stream data by examining the data recorded on a given process in the past (it is similar to the learning process). The discovery of the deviation from standard in the observed stream data enables us to identify that there is something wrong. It does not, however, enable us to name the problem. In the article, we describe the mechanism of the characterization of problem states and the creation of their hierarchies using FCA.

This paper is organized as follows: Section 2 contains an overview of related work. Section 3 explains the method used for evaluation of data. Section 4 describes application of FCA for creating the model of product quality and contain the illustrative example of using the method. Last section 5 concludes the paper.

2 Related Work

The most commonly used methods of monitoring the quality of products include statistical methods [8, 12]. Depending on the type of process the quality control may be carried out on randomly selected products, or it can be performed by an expert on every product. These methods are able to determine the final quality of the product, but are not able to detect errors online during the manufacturing process.

To determine whether a product meets the quality requirements it is, in many cases, necessary to use a system that can determine the quality of the product based on measurements taken during different stages of production. Among these, we can find methods that use regression models and models of neural networks [9, 15]. These methods enable us to predict the quality of the product based on the models we create. These quality control systems are specific to each production, so there are a large number of applications tied to a particular manufacturing process. Our approach is inspired by the previous research, especially in the area of FCA, but we have tried to address several issues of the statistical methods of product quality management in different ways. We did not find any related work focused on product quality management and formal concept analysis. This is the main difference of our approach and previously mentioned related work. Our method was successfully tested in the metallurgical production process where there were several thousand of observed values.

3 Formal Concept Analysis

In our paper, we use Formal concept analysis as a technique for unsupervised clustering. This method helped us to find non-trivial clusters of authors and their keywords. In the next paragraph, we briefly describe Formal Concept Analysis.

Formal concept analysis (FCA) is a general data analysis method based on the lattice theory. FCA was introduced in 1982 by Wille [16]. The basic algorithms for concept lattice computation were published by Ganter in 1984 [4]. More recent publications of these founders can be found in [5, 6, 7]. Carpineto and Romano summarized in [1, 2], both the mathematical and computer scientist’s (with a focus on information retrieval) perspective of the FCA. A good overview of the recent state was written also by Priss [13].

The input data for FCA we will call formal context C , which can be described as $C = (G, M, I)$ - a triplet consisting of a set of objects G and set of attributes M , with I as relation of G and M . The elements of G are defined as objects and the elements of M as attributes of the context.

As an example of context used by FCA method, we have selected five products p_1, \dots, p_5 and five values that were often contained in measurement on the product line. These values were the “temperature inside metallurgical furnace is in bounds - v_1 ”, “speed of the line is in bounds - v_2 ”, “pressure is in bounds - v_3 ”, “chemical composition is OK- v_4 ” and “time of cooling is in bounds - v_5 ”. The relation between products and measured values is shown as a cross in the Table 1

Table 1 Formal context

	value v_1	value v_2	value v_3	value v_4	value v_5
product p_1	×	×	×		×
product p_2	×	×		×	
product p_3	×		×		
product p_4	×			×	×
product p_5		×		×	

For the set $A \subseteq G$ of objects we define A^\uparrow as the set of attributes, common to the objects in A . Correspondingly, for a set $B \subseteq M$ of attributes we define B^\downarrow as the set of objects which have all attributes in B . A formal concept of the context C is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^\uparrow = B$ and $B^\downarrow = A$. The set A is called extent of a concept, while the set B is called intent of a concept. $\mathcal{B}(G, M, I)$ denotes the set of all concepts of context C and forms a complete lattice (so-called Galois lattice). For more details see [6, 7].

4 Make Model of Product Quality on the Product Line

There are a number of values, affecting the final quality, monitored during the production process. These values are very similar during the production of one type

of product, because the production process is set to the produce fault-free products. The manufacturing processes on the production line usually take turns and thus there is historical data available for production model creation.

This historical data can be divided into two groups:

- A set of data corresponding to products that meet the required quality.
- A set of data corresponding to those products that do not meet the required quality.

The stability of a production process brings us advantage of relatively small structure of concept lattice and therefore, take a short time to compute it. Making of model product quality can be done by using historical measurement values. Evaluation of quality product can be done in two phases (learning and classification).

4.1 Pre-processing of Historical Measurement Data

According to the number of different measured values and their ranges, it is necessary to use some method for transformation this many-valued context into one-valued context. The theory around FCA can help us with this task. There is process called conceptual scaling, and we can use some of the standard scales such as “nominal”, “ordinal”, “inter-ordinal”, “bi-ordinal”, etc. scales. As a result of the conceptual scaling we obtain a one-valued context, which is used in the next processing. For detail information of conceptual scaling see [7].

4.2 First Phase (Create Model)

In the first phase (learning) of the quality monitoring system implementation, there are two normalized contexts based on historical measurement data:

- Context containing acceptable states (fault-free product)
- Context containing problem states (fault product)

Although we often work with large amounts of measured data, these contexts are, due to the stability (system requirements), relatively small. Based on these two contexts, there are two conceptual lattices created. These lattices together make the model of manufacturing process.

4.3 Second Phase (Quality Evaluation)

There are three possible scenarios of online product quality monitoring.

Measured values obtained from the production line are mainly as a data stream:

- Comply with any pattern of standard production (lattice of acceptable states)
- Correspond to a known problem (lattice of problem states)

- Are identified as a previously unknown anomaly. In this case, expert assistance is required to assess the values and their impact on quality. Possible evaluations are as follows:
 - Accepted state \Rightarrow model update
 - Problem \Rightarrow problem situations model update

When processing the input data it is possible to monitor the quantity of high quality and low quality products. This piece of information is stored in the form of a counter in the algorithm that will be described later.

4.4 *Some Problems of Used Approach*

During implementing of our approach to the product line in company, we have identified some problems. These problems can be divided into two groups.

- **Contradictory states**
The problem of contradictory states is in our approach solved by the expert. Sometimes is not easily identify where a problem is and only expert, in particular area can choose the right decision. Our final goal for this problem is a replacement of an expert by the expert system.
- **Noise in the measurements**
During the monitoring of the production line there would may occur noise values (empty values, random values) in the measurements. We are working on this problem, and we can solve it by using concept stability and concept approximation during phase of creating the model. For more information see [10].

4.5 *Pseudocode of Quality Evaluation*

Pseudocode of the quality monitoring system behaviour can be described by the following algorithm (Listing 1).

The input parameters of the algorithm – ModelR (meets requirements) and ModelE (error and anomalies) - are conceptual lattices (representing system model) that have been created from historical data of the production process. This data was scaled prior to the model creation [1]. ModelR lattice was created from the reduced context, which contained only those measured values that meet the requirements for the product quality. ModelE lattice was created only from those measured values that did not meet the requirements for the product quality. The vector of measured values is adjusted by conceptual scaling to such values that can be used for inclusion into the conceptual lattice.

Lines 2-7: Using a modified algorithm for incremental update of the lattice, we will try to add an input vector of values V into the ModelR lattice. If the addition of the input vector does not result in the necessity of creation of a new concept, the counter will be updated in the concepts that were changed by adding the vector

Listing 1 Pseudocode of quality evaluation

```

1  procedure PredictionQuality(ModelR , ModelE , V)
2  NewConcept := ModelR . CheckNewConcept (V);
3  If Not (NewConcept) Then
4      ModelR . UpdateCounters (V)
5      ReportQualityFrom (ModelR )
6      Goto End
7  End
8  NewE := ModelE . CheckNewConcept (V);
9  If (NewE = Empty) Then
10     ModelE . UpdateCounters (V)
11     ReportQualityProblemFrom (ModelE)
12     Goto End
13 End
14 Else
15     Answer := AskExpert (V)
16     If (Answer = Error) Then
17         ModelE . AddIncremental (V)
18         ReportQualityProblemFrom (ModelE)
19     Else If (Answer = NewRightState) Then
20         ModelR . AddIncremental (V)
21     End
22 :End

```

data. The operator will be informed of the current product quality. In cases when a new concept is created during the vector addition, the input data represent the state which has not yet appeared in the production history.

Lines 8-13: We try to insert the input vector of values into the ModelE lattice. This lattice contains known historical errors or exceptional conditions in the production when the product quality did not meet the required values. If by addition of the vector into the lattice, there is no new concept, the input vector represents a known error in the production process. In this case, the counters are incremented in those concepts in which the change was made. The operator is then informed of the error in the product quality. If, by addition of the vector into the lattice, there is a new concept created, it is necessary to take an action because the similar state has not yet appeared in the production history. At this moment, we cannot decide whether there is an emergency, for which the product quality meets its requirements, or if there is a production process error.

Lines 15-21: Determining what the status is up to an expert. The expert decides if the status is an emergency that has not yet appeared in the production history while the product quality meets its requirements. In this case, the ModelR lattice is adjusted by the input vector, which enables us to make the model more accurate while maintaining the quality required. Similarly, the ModelE lattice is adjusted in those cases when the expert decides that the monitored product quality does not meet its requirements.

In the next section, we present illustrative example of our approach. Because data from the production line are confidential and too complex, we decided to use simple data based on real measurement in home environment.

4.6 Example of Used Approach

For the purposes of this article, we have created an example that is based on actual measured values of temperatures in two reference rooms of a family house, outside temperature and power consumption necessary for the operation of the boiler for heating the house. The data used in this example include information on the hourly temperatures and power consumption during february 2011¹.

In the period between 02/01/2011 and 02/28/2011 was obtained a total of 671 vectors of hourly measured values. These vectors contain temperature in the bedroom (“Temp 1”), temperature in the nursery (“Temp 2”), outside temperature (“Temp out”), average hourly power consumption (“Consumption”). Examples of measured values are in Table 2.

Table 2 Example of measured data

Date	Hour	Temp 1	Temp 2	Temp out	Consumption
02-2-2011	6:00	17.5	21.1	0.2	10.7
02-18-2011	21:00	18.5	21.5	-2.5	186.1
02-27-2011	23:00	19.5	20.7	-3.9	104.9

The many-valued context was constructed from the measured values of vectors. This context contained 651 rows and five columns. For the individual measured values, there were obtained maximum and minimum values, and they were used in the conceptual scaling. Finally, we got a binary context with 651 rows and 12 attributes. For each of these rows were set property of fault free state.

Retrieved scaled context is reduced in the next step over objects. In the context of the scale, there were many duplications, and these should be removed. The resulting reduced context contains 28 rows and 12 attributes. There are 172 concepts obtained from the context. These concepts form the model of the heating system in the given month. Conceptual lattice is not displayed due to its range. This model is made up solely of measurements that correspond to the heating which is controlled by the controller located in the heating system. There can be interventions to the regulations identified in the historical data – turn off of the heating. We have identified 20 of these manual changes. These measurements correspond to the correct function of the system, but they are exceptional. Therefore, it is necessary to create another model for these manual changes. The model contains their special conditions, and it takes account of the possible error states as well. Capturing of error

¹ Available from <http://radvansky.net/download/mcsd12.zip>

states enables us further analyses their causes. Therefore, there is an attribute added by an expert, which indicates a state as exceptional but not as an error. All previously mentioned structures make up a model of behaviour in the home heating in the month of February 2011 lattice containing error states is depicted in the Fig. 1a. Based on this model, we will further examine the incoming stream of measured values and then decide whether the heating system works fine or not. For further analyses, each concept will be extended for additional information, particularly the number of occurrences of a given concept. (In practice, this could mean a number of products with the same features as were in its production).

We show changes in concept lattice measured values in the data stream in Table 3.

Table 3 Measured errors in the data stream

	Temp 1	Temp 2	Temp out	Consumption
err1	18.3	21.1	-4.2	123.1

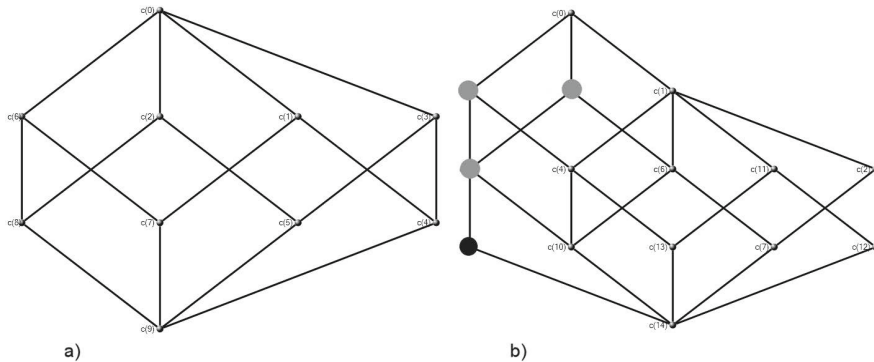


Fig. 1 Changes in the lattice of exceptional states

In the Fig. 1b we can see changes in the lattice of error states (model of anomalies). Black point in the lattice is a newly created concept. This is a symptom of the new situation. The measured values describe the new problem case. Grey circles show newly added concepts created as the extension of the existing concepts.

Once the error states are confirmed, the lattice can be changed into fault behaviour system model. In this case, it would be necessary to merge all objects within the extent into a new object representing the system status. Further analysis of an input measurement process is the same as in the pseudocode of the algorithm showed earlier. Only in the presence of a new system state (error), which generates a new concept in the lattice, it is necessary for the operator to set the corresponding properties of the input data (error/exception).

During the steps of processing the lattice from atoms to the largest element of the lattice, it is possible to identify a common cause of error. This is due to the occurrence of particular attributes along the way. When the production is finished, we gain a model of error and exceptional states that have emerged. Then we can use the model for further repetitive production.

5 Conclusion and Outlook

In this example, we have demonstrated a functional approach to use FCA methods and streaming data in the system management, and production quality management respectively. Although these needs can be addressed by many other methods, proposed solution is suitable to a large number of measured values and a large number of products. In these cases, our solution is able to set the level of quality prediction of particular products online as well as to provide experts with enough information to find potential problems. At the same time, our method enables us to identify the causes of poor quality. The poor quality is not necessarily associated with the individual measured values, but it can also be caused by mutual links between the monitored values, each of which meets the requirements in itself.

Using our solution it is possible to identify both cases. Another considerable benefit of the approach is that experts can obtain a hierarchical view of problem situations and thus gain insight on the quality process. These characteristics make our solution different from the commonly used methods. The solution presented above has been successfully used in practice for metallurgical operation quality monitoring. In the next stage of development, we want to focus on method of approximation of the input vector values, their integration into the lattice and generation of association rules that gives better support for expert's decision making.

Acknowledgements. This project has been realized with the financial support of the Ministry of Industry and Trade of the Czech Republic.

References

1. Carpineto, C., Romano, G.: *Concept Data Analysis. Theory and Application*. John Wiley & Sons, New York (2004)
2. Carpineto, C., Romano, G.: *Using Concept Lattices for Text Retrieval and Mining*. In: Ganter, B., Stumme, G., Wille, R. (eds.) *Formal Concept Analysis. LNCS (LNAI)*, vol. 3626, pp. 161–179. Springer, Heidelberg (2005)
3. Franceschet, M.: *The Role of Conference Publications in CS*. *Communications of the ACM* 53(12) (December 2010)
4. Ganter, B.: *Two Basic Algorithms in Concept Analysis*. In: Kwuida, L., Sertkaya, B. (eds.) *ICFCA 2010. LNCS*, vol. 5986, pp. 312–340. Springer, Heidelberg (2010)
5. Ganter, B., Stumme, G., Wille, R. (eds.): *Formal Concept Analysis. LNCS (LNAI)*, vol. 3626. Springer, Heidelberg (2005)

6. Ganter, B., Wille, R.: Applied Lattice Theory: Formal Concept Analysis. In: Grätzer, G.A. (ed.) *General Lattice Theory*, pp. 592–606. Birkhäuser (1997)
7. Ganter, B., Wille, R.: *Formal Concept Analysis – Mathematical Foundations*. Springer, Berlin (1999)
8. Hald, A.: *Statistical Theory of Sampling Inspection by Attributes*. Academic Press, London (1981)
9. Khosravi, A., Nahavandi, S., Creighton, D.C.: Predicting Amount of Saleable Products Using Neural Network Metamodels of Casthouses. In: ICARCV, pp. 2018–2023. IEEE (2010)
10. Kuznetsov, S.O.: On Stability of a Formal Concept. *Annals of Mathematics and Artificial Intelligence* 49(1), 101–115 (2007)
11. van der Merwe, D., Obiedkov, S., Kourie, D.: AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In: Eklund, P. (ed.) *ICFCA 2004*. LNCS (LNAI), vol. 2961, pp. 372–385. Springer, Heidelberg (2004)
12. Montgomery, D.C.: *Introduction to Statistical Quality Control*. John Wiley & Sons, New York (1985)
13. Priss, U.: Formal Concept Analysis in Information Science. *Annual Review of Information Science and Technology* 40, 521–543 (2006)
14. Roth, C., Obiedkov, S., Kourie, D.: Towards Concise Representation for Taxonomies of Epistemic Communities. In: Yahia, S.B., Nguifo, E.M., Belohlavek, R. (eds.) *CLA 2006*. LNCS (LNAI), vol. 4923, pp. 240–255. Springer, Heidelberg (2008)
15. Shi, G., Zhou, X., Zhang, G.: The Use of Artificial Neural Network Analysis and Multiple Regression for Trap Quality Evaluation: A Case Study of the Northern Kuqa Depression of Tarim Basin in Western China. *Marine and Petroleum Geology* 21(3), 411–420 (2004)
16. Wille, R.: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In: Rival, I. (ed.) *Ordered Sets*, pp. 445–470. Reidel, Dordrecht–Boston (1982)

Soft Competitive Learning for Large Data Sets

Frank-Michael Schleif, Xibin Zhu, and Barbara Hammer

Abstract. Soft competitive learning is an advanced k-means like clustering approach overcoming some severe drawbacks of k-means, like initialization dependence and sticking to local minima. It achieves lower distortion error than k-means and has shown very good performance in the clustering of complex data sets, using various metrics or kernels. While very effective, it does not scale for large data sets which is even more severe in case of kernels, due to a dense prototype model. In this paper, we propose a novel soft-competitive learning algorithm using core-sets, significantly accelerating the original method in practice with natural sparsity. It effectively deals with very large data sets up to multiple million points. Our method provides also an *alternative fast kernelization* of soft-competitive learning. In contrast to many other clustering methods the obtained model is based on only few prototypes and shows natural sparsity. It is the first natural sparse kernelized soft competitive learning approach. Numerical experiments on synthetical and benchmark data sets show the efficiency of the proposed method.

1 Introduction

Clustering algorithms are successful unsupervised machine learning algorithms partitioning a data set into groups. They are able to deal with complex data sets by employing appropriate distance measures. Soft-Competitive-Learning (SCL), an extension of k-means, is a very effective clustering algorithm [10]. It has been successfully applied in different domains like time series prediction [10], image and signal processing [8], bioinformatics and others. In [11, 12] different kernelizations and improvements thereof were proposed, leading to models, competitive to e.g. kernel k-means [15].

Frank-Michael Schleif · Xibin Zhu · Barbara Hammer
CITEC Centre of Excellence, Bielefeld University, 33615 Bielefeld, Germany
e-mail: {fschleif, xzhu, bhammer}@techfak.uni-bielefeld.de

Kernel clustering methods got much attention in the last years [3, 4]. The basic idea of kernel methods is to map the low-dimensional data into a high-dimensional feature space, induced by a kernel function, to obtain linear separability. For SCL this is either done by directly kernelizing the original method [11], its batch variant [7], or by using specific differentiable kernel-functions [13].

For other methods, e.g. kernel k-means, also the support vector description was used [3], referred to as kernel-grower. Kernel grower is based on the classical k-means algorithm and the one-class SVM concept. It maps the data to the kernel-induced feature space and processes the data in an iterative strategy like k-means until a stopping criterion is met. However, instead of computing the centers directly based on the data it computes the smallest sphere enclosing the data by means of the support vector data description (SVDD) [17]. While very effective and flexible from a theoretical point of view it is inefficient for large data sets. Liang et al. [9] proposed to use the core-set concept instead of SVDD in kernel-grower leading to an efficient kernelized core-set clustering with linear complexity in the number of points.

Kernel k-means and its extensions suffer from initialization dependence, which is still subject of research [19]. These issues are fundamental in the k-means algorithm and also kernel-grower and scaled kernel-grower are affected by this.

SCL provides an interesting alternative to achieve less initialization dependent clustering solutions in a k-means like manner [10] but the kernelized approaches of SCL like [11, 13] suffer from its high complexity. Kernel-SCL (KSCL) [11] represents the prototypes or cluster centers by a linear combination of the data points, using a learned coefficient matrix. This coefficient matrix is typically dense such that a lot of data points contribute to the representation of a prototype. Also accelerations of kernel SCL and the introduction of additional sparsity costs in the kernel SCL cost function (see [12]) still do not scale for (very) large data.

In this paper we propose a new algorithm for the kernelization of SCL by means of the core-set technique, which we call core-SCL. The main innovation is to calculate the center, or prototypes of a cluster by means of a small core-set of points which can be efficiently identified by the core-set algorithm. However, the final centers are not directly obtained from the core-set solution like in [9] but by calculating a soft competitive learning solution as detailed in the following. In our algorithm the competitive learning update has much lower complexity, because it is based only on a typically very small set of points defined by the core-set solutions. The overall complexity of our algorithm is still $C \times O(N)$ for a full training procedure, which is the same as for classical SCL but the constant costs C are substantially lower, leading to feasible runtime under practical conditions in contrast to the classical SCL. Core-SCL permits the clustering of up to multiple million of points with linear complexity and is less sensitive to initialization or local minima.

The obtained clustering model is finally based on the adapted core-set information of each cluster. The number of core-set points is typically very small compared to the original cluster size. The underlying coefficient matrix is more sparse compared to KSCL. The cluster centers are determined using these few coefficients.

We evaluate our new approach on synthetic and real life data and compare with different alternative clustering techniques. The paper is organized as follows. In section 2 and 3 we give some preliminary information, followed by a review of kernel soft-competitive learning (kernel-SCL). Section 4 presents the core-SCL algorithm. In section 5 an empirical evaluation on different benchmarking data are shown. We conclude with an outlook and open problems in section 6.

2 Soft Competitive Learning

Let $V = \{\mathbf{v}_1, \dots, \mathbf{v}_N\}$ be a data set with vectors $\mathbf{v}_j \in \mathbb{R}^d$, d denoting the dimensionality, N the number of samples. We call *codebook* the set $W = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$ with each elements $\mathbf{w}_i \in \mathbb{R}^d$ and n as the number of codebook vectors or prototypes. Typically $n \ll N$ and scales in the expected number of clusters. \mathbf{v}_i are represented by one prototype \mathbf{w}_j as explained in more detail subsequently. The prototypes induce a clustering by means of their receptive fields which consist of the points \mathbf{v} for which $d(\mathbf{v}, \mathbf{w}_i) \leq d(\mathbf{v}, \mathbf{w}_j)$ holds for all $j \neq i$, $d(\cdot, \cdot)$ denoting a distance measure, typically the Euclidean distance.

Kernel methods are attractive approaches to analyze complex data which are not linear separable e.g. in the Euclidean domain. The data are mapped using a non-linear transformation function $\Phi : V \rightarrow \mathcal{F}$ into a potentially high-dimensional feature space where a linear separation of the data can be achieved. Typically this is done implicitly using the so called kernel trick [14] and a kernel function.

A kernel function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is implicitly induced by the feature mapping Φ into some possibly high dimensional feature space \mathcal{F} such that

$$\kappa(\mathbf{v}_1, \mathbf{v}_2) = \langle \Phi(\mathbf{v}_1), \Phi(\mathbf{v}_2) \rangle_{\mathcal{F}} \quad (1)$$

holds for all vectors \mathbf{v}_1 and \mathbf{v}_2 , where the inner product in the feature space is considered. Hence κ is positive semi-definite. Using the linearity in the Hilbert-space, we can express dot products of elements of the linear span of Φ in the form $\sum_i \alpha_i \Phi(\mathbf{v}_i)$ and images $\Phi(\mathbf{v})$ via the form $\sum_i \alpha_i \kappa(\mathbf{v}_i, \mathbf{v})$. This property is used in [11], to derive a kernelization of soft competitive learning but also many other prototype based approaches have been extended by kernel concepts [4].

3 The Soft Competitive Learning Network

The SCL algorithm is a type of vector quantizer providing a compact representation of the underlying data distributions [10]. Its goal is to find prototype locations \mathbf{w}_i such that these prototypes represent the data V , distributed according to \mathbb{P} , as accurately as possible, minimizing the energy function:

$$E_{SCL}(\gamma) = \frac{1}{C(\gamma, n)} \sum_{i=1}^N \int \mathbb{P}(\mathbf{v}) \cdot h_{\gamma}(\mathbf{v}_i, \mathbf{W}) \cdot (\mathbf{v} - \mathbf{w}_i)^2 d\mathbf{v} \quad (2)$$

with neighborhood function of Gaussian shape:

$$h_{\gamma}(\mathbf{v}_i, \mathbf{W}) = \exp(-r_i(\mathbf{v}, \mathbf{W})/\gamma) \quad (3)$$

$r_i(\mathbf{v}, \mathbf{W})$ yields the number of prototypes \mathbf{w}_j for which the relation $d(\mathbf{v}, \mathbf{w}_j) \leq d(\mathbf{v}, \mathbf{w}_i)$ is valid, i.e. the winner rank. $C(\gamma, n)$ is a normalization constant depending on the neighborhood range γ . The SCL learning rule is derived by stochastic gradient descent:

$$\Delta \mathbf{w}_i = \varepsilon \cdot h_{\gamma}(\mathbf{v}_i, \mathbf{W}) \cdot (\mathbf{v} - \mathbf{w}_i) \quad (4)$$

with learning rate ε . Typically, the neighborhood range γ is decreased during training.

We now briefly review the main concepts used in Kernelized Soft Competitive Learning (kernel-SCL) proposed in [11]. Kernel-SCL optimizes the same cost function as SCL but with the Euclidean distance substituted by a distance induced by a kernel. Since the feature space is unknown, prototypes are expressed implicitly as linear combination of feature vectors

$$\mathbf{w}_i = \sum_{l=1}^N \alpha_{i,l} \Phi(\mathbf{v}_l) \quad (5)$$

$\alpha_{i,\cdot} \in \mathbb{R}^N$ is the corresponding coefficient vector. The coefficient vector are stored in a matrix $\Gamma \in \mathbb{R}^{n \times N}$. Distance in feature space for $\Phi(\mathbf{v}_j)$ and \mathbf{w}_i is computed as:

$$d_{i,j}^2 = \|\Phi(\mathbf{v}_j) - \mathbf{w}_i\|^2 = \|\Phi(\mathbf{v}_j) - \sum_{l=1}^N \alpha_{i,l} \Phi(\mathbf{v}_l)\|^2 \quad (6)$$

$$= k(\mathbf{v}_j, \mathbf{v}_j) - 2 \sum_{l=1}^N k(\mathbf{v}_j, \mathbf{v}_l) \cdot \alpha_{i,l} + \sum_{s,t=1}^N k(\mathbf{v}_s, \mathbf{v}_t) \cdot \alpha_{i,s} \alpha_{i,t} \quad (7)$$

The update rules of SCL can be modified by substituting the Euclidean distance by the formula (6) and taking derivatives with respect to the coefficients $\alpha_{i,l}$. Further, substituting the prototypes by linear combinations, the adaptation rule of the Γ matrix becomes

$$\alpha_{jl} := \alpha_{jl}(1 - \eta h_{\sigma}(r_{ij})) \text{ if } l \neq i \quad (8)$$

$$\alpha_{ji} := \alpha_{ji}(1 - \eta h_{\sigma}(r_{ij})) + \eta h_{\sigma}(r_{ij}) \quad (9)$$

For a Gram matrix, we can w.l.o.g. restrict the coefficients such that prototypes are contained in the convex hull of the data points in the feature space, i.e. the coefficients are non-negative and sum up to one. It should be noted that the kernel-SCL and its batch variant [7] have a complexity of $O(N^2)$, each. Both methods represent the prototypes by linear combinations of the data points. The coefficients $\alpha_{i,j}$

are stored in the matrix Γ which is typically dense. Therefore the distance calculations between a prototype and a datapoint requires almost all points due to the reconstruction of the prototype via the linear combination.

4 Core Soft Competitive Learning

The notion of core-sets appears in solving the approximate minimum enclosing ball (MEB) problem in computational geometry [11].

Definition 1 (Minimum enclosing ball problem). Let $\{\mathbf{v}_1, \dots, \mathbf{v}_m\} \in \mathbb{R}^d$ a set of points. The objective is to find a minimum enclosing ball with radius R and center point \mathbf{c} such that $\|\mathbf{c} - \mathbf{v}_i\|^2 \leq R^2 \forall i$. Hence $B(\mathbf{c}, R) = \{\mathbf{v} | R \geq \|\mathbf{c} - \mathbf{v}\|, \mathbf{v} \in V\}$.

As shown in [18], the MEB problem can be equivalently express as a quadratic dual optimization problem:

$$MEB = \min_{\alpha_i \geq 0, \sum \alpha_i = 0} \alpha K \alpha^\top - \sum_i \alpha_i K(i, i)$$

with K the kernel matrix defined on V

and \mathbf{c} and \mathbf{v} represented in a kernel space as shown before. The radius R is obtained by solving $R = \sum_i \alpha_i K(i, i) - \alpha K \alpha^\top$. The center is used only in the distance calculation which can be expressed using the kernel trick and the linear combination of \mathbf{c} based on the obtained α -vector, $\mathbf{c} = \sum_{i=1} \alpha_i \phi(\mathbf{v}_i)$ as shown for kernel-SCL.

Definition 2 (Core set). Let $S = \{\mathbf{v}_1, \dots, \mathbf{v}_k\} \subseteq V$, then $Q \subset S$ is a core set, if $S \subset B(\mathbf{c}, (1 + \varepsilon)R)$ and $B(\mathbf{c}, R) = MEB(Q)$.

An encouraging property of core-sets is that the number of elements in it is independent of the data dimensionality and size [11]. The main concepts involved in core-sets are illustrated in Figure 1 (left).

The core-SCL algorithm can be calculated in the Euclidean space or using a kernel-induced feature space. In the following we will present core-SCL for the more generic case of arbitrary kernels. The core-SCL clustering is initialized by specifying the number of prototypes n and an initial coefficient matrix $\Gamma = n \times N$. We initialize the matrix Γ randomly such that for each row (α_i) two entries have a value of 0.5. Further a learning rate and neighborhood cooperation is initialized equivalently to the standard SCL and a copy of the Γ -matrix, $\hat{\Gamma}$ is stored for later use in the update. The α_i encode the prototype positions using Equation (5). In the learning phase the Γ -matrix is adapted such that the prototype positions are optimized following the optimization scheme of SCL. The algorithm iterates the following steps:

1. calculate the receptive fields using Eq. (6) with $\hat{\Gamma}$
2. if the receptive field has not changed, continue with step 4
3. calculate the core-set for each receptive field using the algorithm of [11]

4. calculate distances between \mathbf{W} and all *core-set points* using Eq. (6)
5. calculate the corresponding neighborhood function using Eq. (3)
6. apply Eq. (8) to Γ using $\hat{\Gamma}$ with respect to the core-set points
7. store Γ as $\hat{\Gamma}$, continue with step 3

Like in [9] we limit the influence of potential outliers in the current approximation of the receptive field, by modifying the step 3 in the prior algorithm such that a fraction τ of the points can be left out during the core-set approximation. Further we use probabilistic speedup [18]. The overall algorithm stops if the receptive fields did not change for a number of iterations or an upper number of cycles is reached. In each cycle the learning rate and neighborhood range is adapted in the same manner as for kernel-SCL.

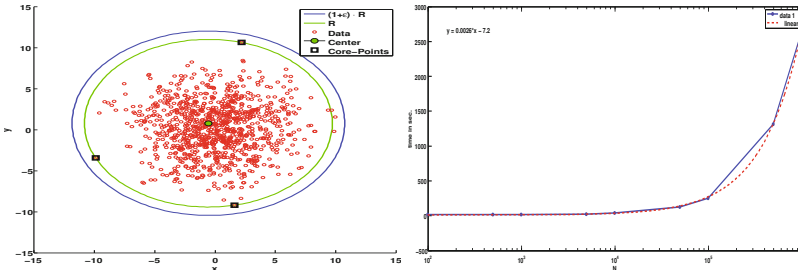


Fig. 1 (Left) Core-Set schema: Minimum enclosing ball solution of a set of data points. The core-set consists of the points indicated by squares, implying a minimum enclosing ball of radius R and its approximation with a radius $(1 + \varepsilon) \cdot R$ (outer circle). The outer circle includes all points of the data within a tolerance of $(1 + \varepsilon) \cdot R$. In this example 1000 points are described by a circle defined upon 3 core-set points. (Right) Runtime analysis of core-SCL varying N on a checkerboard data set. Note that the x-axis uses log-scale. A linear fit (red) agrees well with the expected complexity of $O(N)$.

4.1 Runtime Analysis

The core-SCL calculates prototype positions for a constant number of cycles. In each cycle the receptive fields have to be calculated, with $n \times N$ distance calculations and n sorting operations to determine the winner. Hence step 3 has a complexity of $O(N)$. This is a substantial difference to kernel-SCL or batch kernel-SCL where these calculations have $O(N^2)$ complexity. The quadratic complexity of these algorithms also remains if a k -approximation of the Γ matrix is done (e.g. only the k largest alpha values are kept), because this can only be applied after an initial number of steps, typically 10 cycles or in the final model to ease later interpretation.

The core set calculation (step 3) of each receptive field has worst case runtime complexity of $O(N)$. This is because in each core-set iteration a new core-set vector is added until the MEB is found, which requires the test $R \geq \|c - \mathbf{v}\|$, $\mathbf{v} \in V$ for each point in the receptive field which is not yet within the MEB. Employing probabilistic speedup [18], this complexity can be substantially reduced in practice, checking

only a random subset of the points. As discussed in [18], it is sufficient to select one (or multiple) random subsets of 59 points $\mathbf{v} \in S$, with some moderate assumptions on the distance distribution. As shown in [16] by using a small random sample S' from S the closest point obtained from S' is with probability 95% among the closest 5% of points from the whole S .

The distance and rank calculations between the prototypes and the core-set points is done in $O(n \times Z)$ including sorting in $O(\log n)$, where Z is the size of all points which are core-set points. The obtained ranks are used to update Γ using $\hat{\Gamma}$.

Note that the number of core-set points per MEB calculation is bounded by $O(\frac{1}{\epsilon^2})$ [1] and hence remains small with respect to each cluster. Due to the small number of core-set points per cluster we have $Z \ll N$. Hence we have a complexity of $O(N)$ for the steps [4] to [6].

Summarized, the overall complexity is linear $O(N)$ in N , since $n \ll N$. This is also reflected by the runtime analysis shown in Figure [1] (right).

4.2 Memory Complexity Analysis

The consumed memory is mainly determined by the size of the kernel-matrix $K = N \times N$ and the coefficient-matrix Γ which is $n \times N$. The full kernel matrix does not need to be calculated but only a sub-part is necessary in each iteration. In core-SCL the distance between the n prototypes and all N datapoints needs to be calculated. At this step the corresponding kernel sub-matrix has to be calculated. It consist of the rows of the non-vanishing α_{ij} for a row i . And relates directly to the identified core-sets indices causing the non-vanishing alphas. If we define $S_{all} = \{S_1, \dots, S_n\}$ as the unique list of core-set index sets for each cluster, the corresponding sub-matrix in the calculation of the receptive field i is roughly $|S_i| \times N$, which is typically much smaller than N^2 and hence remains linear. The number of core-set points in an MEB calculation is bounded by $O(\frac{1}{\epsilon^2})$ [1]. This calculation can potentially be further simplified by using the Nyström approximation as shown e.g. in [12]. The Γ matrix has a complexity of $n \times N$, which can probably be reduced further using appropriate storage-classes due to the sparsity of Γ . Accordingly, the memory complexity is roughly linear in $O(N)$.

5 Experiments

In the following we show the efficiency of core-SCL on some artificial and real life data sets and compare with results reported in [9]. First we start with a checker-board simulation of 3×3 fields. Each cluster is Gaussian with clear separation between the means and a small variance, causing small overlap. The 9 prototypes are initialized randomly in the data and the objective is to position each prototype in a cluster. In Figure [2] we show a run of standard kernel k-means and core-SCL on the

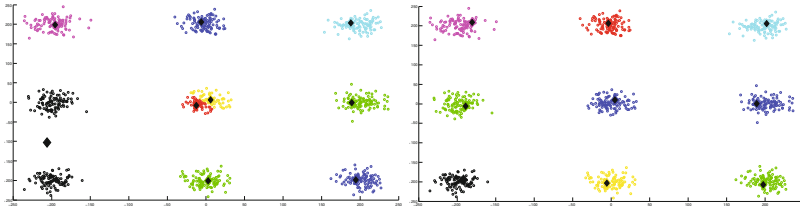


Fig. 2 Results for the checkerboard data with 100 points per cloud, black markers indicated the prototypes. Left: kmeans - obviously fails to point one prototype per cluster due to initialization problems and dead-lock situations. Right: core-scl, each cluster is represented by a prototype.

checker-board data using a linear kernel. One observes a classical problem of kmeans which is efficiently overcome with the use of soft-competitive learning. K-means sticks in local minima and is unable to obtain a reasonable solution. As a further example we use the classical ring data set which consist of two rings which are not linearly separable. Each ring has 4000 points with added Gaussian noise in $N(0, 0.25)$. The core-SCL needs 10 steps and an RBF kernel with $\sigma = 0.5$ to separate the two rings as shown in Figure 3. In a 10-fold crossvalidation core-SCL was always able to obtain a perfect clustering, 0%-error and the mean sparsity of the model was around 92%, hence ≈ 300 points have been used to describe the prototypes in each model. Also scaled kernel-grower achieved a perfect clustering for these data and the sparsity was even better using only ≈ 11 points per model to represent the prototypes.

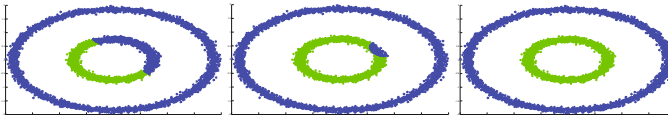


Fig. 3 Core-SCL with 2 prototypes applied to ring data (inner - circles, outer - diamond). Perfect post-labeling at iteration 10 with the inner ring, separated from the outer ring.

In the following we show the effectiveness of core-SCL for two real life, standard data sets taken from the UCI machine learning repository [2] and compare to different clustering approaches. For comparison we consider, affinity propagation, a generic clustering approach based on a factor-graph model [6]. It provides natural sparsity, since the model consists of exemplars in the original data space, equivalent to the number of requested clusters. However AP does not scale for larger data sets and even in the range of about a thousand data points the algorithm is already slow. Additionally we consider a batch variant of K-SCL [7]. In the batch version all data points are processed in one step. This is very efficient, but can not be done for extremely large data sets ($N \gg 10000$) without approximation or sub-sampling.

Table 1 Mean classification accuracy (\pm standard dev.) of core-SCL to other methods within a 5-fold cross-validation. The runtime is given in seconds. The dual costs are given for the test data averaged over all runs. The sparsity is the percentage of points used to reconstruct all prototypes.

	core-SCL	AP	K-SCL (batch)
Landsat			
accuracy	86.00% \pm 1.3	86.3% \pm 0.2	86.1% \pm 0.2
runtime	126.43	1478	329
q-error	1704	1568	1545
sparsity	0.11%	–	3.6%
Phoneme			
accuracy	84.05% \pm 1.8	87.1% \pm 0.2	87.9% \pm 0.3
runtime	92.41	1237	104
q-error	2656	2578	2487
sparsity	0.23%	–	3.6%
Spam			
accuracy	82.29% \pm 4.4	76.00% \pm 4.0	84.35% \pm 1.7
runtime	13.92	304	77
q-error	420.26	606	588.25
sparsity	0.4%	–	100%

Table 2 Mean classification accuracy (\pm standard dev.) of core-SCL for the large data within a 5-fold cross-validation. The runtime is given in seconds. The dual costs are given for the test data averaged over all runs. The sparsity is the percentage of points used to reconstruct all prototypes. AP and K-SCL did not get feasible runtimes.

	Checker	Intrusion
accuracy	100 \pm %0	93.84% \pm 0.21
runtime	107.63	434.50
q-error	1334	15267
sparsity	0.03%	$1e^{-7}\%$

We consider the Landsat satellite data with 36 dimensions, 6 classes, and 6435 samples, the Phoneme data with 20 dimensions, 13 classes, and 3656 samples and the spam dataset with two classes, 57 dimensions and 4601 samples. The results are given in Table 1.

To show the effectiveness of our approach also for large data sets we consider a large version of the checkerboard data, as described above, but with \approx 1 million points and the KDD-intrusion detection taken from [18] with training data of \approx 5 million points. It contains connection records of network traffic. The task is to separate normal connections from attacks on the preprocessed raw data as detailed in [18]. The results for these large scale experiments are shown in Table 2 for core-SCL only, since the other methods took too long for the complete experiments.

¹ The number of examples in AP is the same as the number of cluster, hence the AP model is always the most sparse model possible.

In all experiments K-SCL and core-SCL are initialized randomly. For AP we used the standard settings described in [6]. Experiments run until convergence, with an upper limit of 100 cycles. We use the standard linear kernel and the ELM kernel, a defacto parameter-free, optimal RBF kernel as discussed in [5]. The number of clusters and used kernel are: LANDSAT (100,linear), PHONEME (100,linear), SPAM (2,ELM), CHECKER (9,linear), INTRUSION (2,linear).

6 Conclusions

We proposed a novel kernelization of Soft Competitive Learning employing core-sets. The approach automatically leads to sparse models with low memory consumption compared to the traditional kernel-SCL. The results show that the runtime could be substantially improved using core-sets, avoiding the consecutive updates for all data points per iteration as in the traditional online kernel-SCL, but also compared to batch kernel-SCL, for larger data sets. The algorithm is similarly efficient with respect to the dual-quantization error and post-labeling accuracy compared to the other methods. While AP and K-SCL cannot be used to analyze very large data sets, core-SCL can be easily applied for such data.

Acknowledgements. This work has been supported by the German Res. Found. (DFG), HA2719/4-1 (Relevance Learning for Temporal Neural Maps) and in the frame of the centre of excellence 'Cognitive Interaction Technologies'.

References

1. Badoiu, M., Har-Peled, S., Indyk, P.: Approximate clustering via core-sets. In: STOC, pp. 250–257 (2002)
2. Blake, C., Merz, C.: UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA (1998), <http://www.ics.uci.edu/~mlearn/MLRepository.html>
3. Camastra, F., Verri, A.: A Novel Kernel Method for Clustering. IEEE TPAMI 27(5), 801–805 (2005)
4. Filippone, M., Camastra, F., Massulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. Pattern Recognition 41, 176–190 (2008)
5. Frénay, B., Verleysen, M.: Parameter-insensitive kernel in extreme learning for non-linear support vector regression. Neurocomputing 74(16), 2526–2531 (2011)
6. Frey, B., Dueck, D.: Clustering by message passing between data points. Science 315, 972–976 (2007)
7. Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity data sets. Neural Computation 22(9), 2229–2284 (2010)
8. Labusch, K., Barth, E., Martinetz, T.: Soft-competitive learning of sparse codes and its application to image reconstruction. Neurocomputing 74(9), 1418–1428 (2011)
9. Liang, C., Xiao-Ming, D., Sui-Wu, Z., Yong-Qing, W.: Scaling up kernel grower clustering method for large data sets via core-sets. Acta Automatica Sinica 34(3), 376–382 (2008)

10. Martinetz, T., Berkovich, S., Schulten, K.: Neural Gas Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE Transactions on Neural Networks* 4(4), 558–569 (1993)
11. Qin, A.K., Suganthan, P.N.: A novel kernel prototype-based learning algorithm. In: *Proc. of ICPR 2004*, pp. 2621–2624 (2004)
12. Schleif, F.M., Villmann, T., Hammer, B., Schneider, P.: Efficient kernelized prototype-based classification. *Journal of Neural Systems* 21(6), 443–457 (2011)
13. Schleif, F.-M., Villmann, T., Hammer, B., Schneider, P., Biehl, M.: Generalized Derivative Based Kernelized Learning Vector Quantization. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) *IDEAL 2010. LNCS*, vol. 6283, pp. 21–28. Springer, Heidelberg (2010)
14. Schölkopf, B., Smola, A.J.: *Learning with Kernels*. MIT Press (2002)
15. Schölkopf, B., Smola, A.J., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10(5), 1299–1319 (1998)
16. Smola, A.J., Schölkopf, B.: Sparse greedy matrix approximation for machine learning. In: Langley, P. (ed.) *ICML*, pp. 911–918. Morgan Kaufmann (2000)
17. Tax, D.M.J., Duin, R.P.W.: Support vector domain description. *Pattern Recognition Letters* 20(11-13), 1191–1199 (1999)
18. Tsang, I.W., Kwok, J.T., Cheung, P.M.: Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research* 6, 363–392 (2005)
19. Tzortzis, G., Likas, A.: The global kernel k-means clustering algorithm. In: *IJCNN*, pp. 1977–1984. IEEE (2008)

Enhancing Concept Drift Detection with Simulated Recurrence

Piotr Sobolewski and Michał Woźniak

Abstract. This paper focuses on the concept drift detection and proposes how to extend the functionality of a statistical concept drift detector for unlabeled observations. For those algorithms the previously developed approach so-called simulated recurrence is implemented as a separate module. It provides information regarding the possible data distribution after concept drift detection. The proposed approaches were compared with five detection algorithms on the basis of computer experiments which were carried out on the UCI benchmark datasets.

1 Introduction

The data stream classification task becomes complicated when classification rules are changing. Such phenomenon is called concept drift and it is observed in a numerous real-life scenarios, such as monitoring customer shopping preferences, analyzing internet web browsing trends, predicting levels of stock market indexes, localization systems, credit card fraud detection or flight simulation. When concept drift occurs, performance of an unprepared classification system may decrease, therefore such threat requires special handling.

In general, approaches to cope with concept drift fit in one of the two categories [4]:

- Approaches, which adapt a learner at regular intervals without considering whether changes have really occurred,
- Approaches, which detect concept changes firstly, and then a learner is adapted to these changes.

Adapting the learner is a part of an incremental learning approach. Depending on the type of an used learner, the model is either updated (e.g., neural networks or

Piotr Sobolewski · Michał Woźniak

Department of Systems and Computer Networks, Faculty of Electronics,
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: [piotr.sobolewski,michal.wozniak}@pwr.wroc.pl](mailto:{piotr.sobolewski,michal.wozniak}@pwr.wroc.pl)

traditional decision trees) or needs to be partially or completely rebuilt (as CVFDT algorithm [7]).

In this paper we will focus on the second category, where detector and classifier can be designed separately. Many detection algorithms are using knowledge of object labels after the classification in order to detect concept drift, however as pointed out in [13], such approach does not fit in the real scenarios. In general, concept drift detection algorithms can be divided into three types, depending on the assumption about the amount of costly knowledge regarding the true class labels available for the algorithm, namely:

- Supervised algorithms – assuming access to classification performance measures or true class labels, detecting concept drift on the basis of classifier’s accuracy or analysis of class distributions – although an access to this knowledge is often very expensive and in many practical cases it is impossible to label data e.g., because objects are coming very fast,
- Semi-supervised algorithms – assuming limited access to classification performance measures or true class labels, also detecting concept drift on the basis of the properties of data when such knowledge is not available – a more “rigorous” approach, taking into account the cost of labeling, a flag example in this category is active learning [5], which selects the samples for labeling
- Non-supervised algorithms – assuming no access to classification performance measures or true class labels, basing only on the properties of data, detecting concept drift on basis of attribute value distribution, cluster memberships or classifier’s support levels – after detecting concept drift, usually the labels or knowledge about classification error is still necessary to train a new classification model.

In this paper, we explore the possibilities of detecting concept drift in data streams without supervision. It is worth noting, that this approach has some limitations. Let us consider a situation of so called the real concept drift, when the drift is not visible in the data distribution, but it affects the distribution of labels only. In such situation it is impossible to detect concept drift solely by analyzing the data properties.

2 Related Works

The task of concept drift detection basing on the properties of data can be approached as a multivariate statistical test which checks if a window of data drawn from the data stream comes from the same distribution as the reference dataset. For this purpose, a statistical two-sample test can be used, however parametric tests such as a T2 statistic [6] assume a specific distribution, which might not be a correct approach in the real data case, as the samples may include records from several classes, each described by a different distribution. Also, as the data may not arise from any of the known standard distributions, non parametric tests are more suited for this task. Examples of such tests include:

- **CNF Density Estimation Test [2]**
Approach introduced in [2], describes the data by vectors of binary features, assigned by discretizing attributes into sets of bins. It then creates a set of Boolean attributes A , which “covers” all of the examples in the reference set of data X , meaning that each “true” feature in set A is the same as in at least one of the vectors describing the data points in X . Next, another set of data \bar{X} is drawn from the same distribution as the data in X , represented as binary vectors, and compared to set A , by calculating parameter c_i for each example x_i in \bar{X} , which is measured by counting the number of clauses in set A , which do not “cover” x_i . When a data window \underline{X} is tested to check if it comes from the same distribution as X , a sequence of parameters c_i is measured for all data samples in the window and compared with the sequence of c_i ’s obtained by comparing the distributions of data in X and \bar{X} by applying a Matt-Whitney test. If the difference is insignificant, all data is considered to come from the same distribution, otherwise a difference in distributions is detected and a drift in concept is signaled.
- **The Wald-Wolfowitz Test [3]**
The multivariate version of the Wald-Wolfowitz test [3] constructs a complete graph, with examples as vertices and distances between them as edges. Graph is then transformed into a forest and a test statistic is computed basing on the amount of trees.

As benchmark test statistics, we also evaluate three popular univariate tests, suggested in [13]:

- Two-sample Kolmogorov-Smirnov test,
- Wilcoxon rank sum test,
- Two-sample T-test.

We choose to generalize the univariate statistic tests to multivariate versions [9] by performing the test over each dimension (for every attribute) and signal concept drift if the null hypothesis is rejected for any one of the dimensions.

All of the mentioned above concept drift detection methods comply with the preliminary assumptions about non-supervised detection and independence from the data classification module. Also, as the tests are non-parametric, they can be evaluated on any scenario without pre-setting parameters.

3 Simulated Recurrence

Simulated recurrence was introduced firstly as an extension to classification algorithms coping with recurring concept drift in [10]. Recurring concept drift facilitates an assumption, that the concepts may recur in the data stream, namely after the concept shift, the data may start to follow the same models as were already observed in the past. When it happens, a classification system can use the previously gathered knowledge regarding this concept and quickly adjust the classification rules. The aim of simulated recurrence approach is to re-create such situation in a

non-recurring concept drift scenario. The recurring concept is replaced with an artificially simulated concept. The concept data is generated on the basis of knowledge about the model which is available in advance.

In our previous works [10] [11], simulated recurrence has been used to improve the overall efficiency of the classification systems coping with concept drift, by decreasing the error-rate. The approach has obtained promising results, improving the performance of classification systems by around 25%. The main contribution of this paper compared to our previous works is a completely novel use of simulated recurrence. Instead of improving the classification module's accuracy, we use it for creating an additional module, which enriches the signal produced by the detection module by adding an information regarding the possible data distribution parameters after concept drift occurrence. This information may be used for improving the process of updating a classification model. Two methods for simulating recurrence are evaluated, described in more details in section 4.2.

In the classical detection system, the aim is to determine whether the data window D drawn from the data stream comes from the same distribution as a reference dataset R by analyzing the p -value outputs of the statistical tests. If the null hypothesis is rejected, the data is considered to arise from a different distribution and concept drift is signaled. Such binary detection does not carry much valuable information for the classification module. The use of simulated recurrence may extend the functionality of a binary detector by including the information about the possible data distribution in the new concept. The simulated concept data may be used for training the classification module, decreasing the effort required to tune to the new concept. Implementation of simulated recurrence adds a new block to the two-step detection-classification process, as shown in Fig. 1.



Fig. 1 Detection-identification-classification process with simulated recurrence

After concept drift detection, the data window D together with obtained p -value D is passed to the Simulated Recurrence module, where it is tested with null hypothesis whether it follows the same distribution as the data in the artificial concept dataset A . If the obtained p -value SR is lower or equal to the p -value D , the information about concept is not included in the detection output and the classification system needs to use other mechanisms in order to tune to the new concept. Otherwise, the dataset A is passed to the Classification module as a schema representing a possible data distribution in the new concept. In the case of a univariate detector, instead of the whole window D , only the attribute vector, for which concept drift has been detected is passed to the Simulated Recurrence module.

Notations:

- D - data window,
- R - reference dataset,
- A - artificial concept dataset, created with simulated recurrence.

Algorithm:

1. Pass D to the Detection module,
2. Test null hypothesis about distributions of data in D and data in R (compute p -value D),
3. IF null hypothesis rejected (concept drift detected),
 - a) Pass D and p -value D to the Simulated Recurrence module,
 - b) Test null hypothesis about distributions of data in D and data in A (compute p -value SR),
 - c) IF p -value SR \leq p -value D,
 - i. Classification module builds a new classification model (needs requesting the class labels),
 - ii. Return to 1.
 - d) ELSE
 - i. Pass A to the Classification module,
 - ii. Classification module builds a new model on the basis of A (no need for requesting class labels),
 - iii. Return to 1.
4. ELSE
 - a. Return to 1.

Fig. 2 Pseudo-code of concept identification with simulated recurrence

The Simulated Recurrence module may also serve as a validator for the detector's decisions. If the system has the knowledge about all possible concepts, it can simulate all the scenarios via simulated recurrence. Then, obtaining a p -value SR lower or equal to the p -value D on the level of Simulated Recurrence module may be a signal that the Detection module has made an incorrect decision and it might require re-validation with different mechanisms. This functionality may also be extended to validating all detector's decisions, also the negative ones, however we leave this analysis for future research, outside the scope of this paper.

4 Experimental Investigation

4.1 Data Description

The scenarios are based on 22 UCI datasets [1]. In order to prepare a concept drifting environment, we follow the approach pioneered in [12] and deployed in [2]. The method can be described in two steps, first the data in dataset is ordered by classes

from the most populated to the least populated, next class labels are removed and the data originating from each class is treated as data originating from different concepts. The data from the most populated class forms a reference dataset R and the data from the second most populated class forms a concept dataset, C . As in [2], we compare only the first two classes. In order to evaluate the false-positive detections, the dataset R is divided to a reference dataset R_1 and evaluation dataset R_2 .

4.2 Simulating Recurrence

We evaluate two methods for the simulating recurrence. In the first case, only the knowledge about the mean values of the data attributes in the new concept is available and the distribution is copied directly from the reference dataset. In the second case, also the variances of attribute values are known and the data is generated on the basis of these parameters following the normal distribution, not considering the distribution of data in the reference dataset. Both methods are described in more details below:

Scenario 1 – basing on the knowledge about the means of all attributes of samples in the new concept, $c\mu$, for each observation r_i of all n observations in the reference dataset R with m attributes, an artificial observation a_i is generated by moving the value of j -th attribute by the amount specified by the difference in means for the new concept, $c\mu_j$, and the reference concept, $r\mu_j$:

$$a_{i,j} = r_{i,j} + (c\mu_j - r\mu_j), \quad (1)$$

Scenario 2 – basing on the knowledge about the means and variances of the attribute values in the new concept, $c\mu$ and $c\sigma^2$, n artificial observations are generated with m attributes, each following a normal distribution described by the mean $c\mu_j$ and variance $c\sigma_j^2$:

$$a_{i,j} = N(c\mu_j, c\sigma_j^2), \quad (2)$$

In the second scenario, we are not restricted by the amount of artificial data, which we can generate, however for the experiments, we generate the same number of samples, as in the reference dataset R . Also, it is worth noting that the distributions are kept only within the bounds of single attributes, as the method does not consider relations between the different attributes. In case of highly multi-dimensional data, this method may suffer a severe decrease of efficiency.

4.3 Evaluation Method

In order to evaluate the efficiency of tested methods, we run 100 test cases for each dataset and validate the statistical significance of the obtained results by comparing the scores with a two paired t-test. A single test case consists of one loop of the

algorithm described in Fig. 2 with *window size* D set to 10 samples and the test significance level *p-value* set to 5. During the experiments we used:

- Specificity - percentage of false signals filtered, dependent from the amount of the false-positive detections of Detection module,
- Sensitivity - percentage of correctly identified concepts, based on the correct detections of the Detection module.

4.4 Results

The main goal of the experiments is to evaluate the two methods for simulating recurrence in the manner of sensitivity and specificity, depending on the true-negative or false-positive detections. After each incorrect (false-positive) detection from the Detection module, the Simulated Recurrence module is forced to cope with a false signal, evaluating the sensitivity score. When the detection is correct (true-negative), the Simulated Recurrence module evaluates the specificity score, as the concept should be identified as statistically closer to the data window.

The results are presented in Table 1. It contains the percentage sensitivity and specificity scores of the detection and simulated recurrence modules. First let us summarize the performance of the Detection modules. The results show, that our method of generalizing univariate tests to multivariate detectors makes them over-sensitive, correctly detecting 100% of concept-drifted data windows for almost every scenario, however also making many false-positive errors. The least sensitive univariate test is Kolmogorov-Smirnov and the most sensitive is the Wilcoxon rank sum. The least sensitive overall is the CNF test, what also finds reflection in the lowest number of correct detections overall. A very poor performance with false-positive detections is achieved by the multivariate Wold-Wolfowitz test, which however achieves slightly better scores in correct detections, than the other multivariate test.

Simulated Recurrence achieves best results with the CNF test detector. The first method for simulating recurrence makes almost no false-positive errors and achieves 100% correct concept identifications for most of the scenarios. It shows, that if the method is combined with a reliable and accurate detector, it can achieve really good results. Contrary, although the second method achieves similar scores of correct detections, it achieves a very weak sensitivity score, making a lot more true-negative errors, meaning that it is not suited for CNF test algorithm. For other algorithms, both methods achieve similar performance, which is rather poor in case of the univariate T-test and Wilcoxon rank sum tests.

An interesting case is the scenario with the multivariate version of Wald-Wolfowitz test. The Detection module makes many false-positive detections, however the Simulated Recurrence module does not follow, achieving 100% sensitivity rate for most of the scenarios. It means, that if the system had knowledge about all possible concepts, the Simulated Recurrence module could serve as a filter for

Table 1 Sensitivity and Specificity scores of the Detection and Simulated Recurrence modules

Dataset	CNF		WW		KS		T2		WCX										
	Spec. Sens.		Spec. Sens.		Spec. Sens.		Spec. Sens.		Spec. Sens.										
	(detector)		(detector)		(detector)		(detector)		(detector)										
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2									
(sim. rec.)		(sim. rec.)		(sim. rec.)		(sim. rec.)		(sim. rec.)											
balance-scale	95	97	47	100	98	95	87	100	86	100									
	100	20	100	100	100	100	50	50	100	100	31	31	100	100	36	36	100	100	
breast-w	93	73	52	100	99	100	67	100	65	100									
	100	0	100	100	100	100	100	100	88	67	100	100	100	100	83	100	100		
car	94	37	54	86	94	100	73	91	72	95									
	100	0	100	100	61	79	88	89	34	100	0	70	52	41	100	100	50	58	100
credit-a	96	5	49	86	78	100	50	100	55	100									
	100	0	100	100	100	0	11	50	46	87	100	38	28	100	100	45	29	100	100
credit-g	96	8	51	54	81	62	44	97	41	96									
	0	0	100	100	82	80	32	45	64	74	38	67	34	33	98	99	43	21	95
haberman	97	1	51	81	89	57	85	66	84	71									
	100	0	100	100	96	86	51	83	46	64	65	92	34	34	100	100	50	50	95
heart-c	89	29	56	66	79	66	79	88	60	99	60	99							
	82	10	100	100	50	80	85	84	39	43	71	89	35	40	100	100	45	30	100
heart-statlog	94	52	52	82	85	100	59	100	61	100									
	100	0	100	100	67	78	86	94	60	60	93	100	42	49	100	100	67	42	100
ionosphere	100	0	45	99	46	100	57	100	47	100									
	0	0	0	0	100	57	90	98	32	30	100	100	33	14	100	100	34	21	100
kr-vs-kp	100	0	46	97	91	72	24	100	24	100									
	0	0	0	0	93	100	99	94	78	100	0	57	27	31	100	100	65	14	100
letter	98	71	54	100	87	100	62	100	65	100									
	0	0	100	100	100	100	100	70	85	100	100	43	48	100	100	29	43	100	100
mfeat-morph	100	100	64	100	84	100	86	100	73	100									
	0	0	4	100	100	100	100	94	94	100	100	15	22	100	100	97	97	100	100
nursery	100	100	49	100	95	100	72	100	72	100									
	0	0	100	100	100	100	100	80	100	99	100	36	50	40	100	36	83	87	100
optdigits	100	0	53	100	66	100	17	100	19	100									
	0	0	0	0	100	100	100	42	45	100	100	37	33	100	100	57	30	100	100
page-blocks	96	1	55	100	74	100	76	100	78	100									
	100	0	100	100	100	100	46	50	62	62	100	100	80	21	100	100	46	23	100
pendigits	95	100	57	100	78	100	75	100	74	100									
	100	100	100	100	100	100	100	69	69	100	100	60	64	100	100	74	74	100	100
pima-diabetes	96	5	47	93	74	97	73	99	73	99									
	100	0	100	100	97	91	34	63	54	50	100	100	38	26	100	100	45	34	100
tic-tac-toe	98	12	53	83	89	23	65	51	66	56									
	100	0	100	100	47	90	61	73	82	100	0	74	29	32	97	97	30	30	97
vehicle	100	100	53	100	70	100	63	99	65	99									
	0	0	100	100	100	98	42	65	30	40	99	100	22	41	100	100	46	38	100
vote	98	17	49	100	98	100	59	100	60	100									
	100	100	100	100	100	100	100	0	0	0	100	42	52	100	100	40	48	100	100
waveform	99	59	53	100	33	100	42	100	35	100									
	0	0	100	100	100	96	98	59	62	100	100	56	50	100	100	60	65	100	100
yeast	100	0	58	76	83	59	70	71	66	73									
	0	0	0	0	91	91	61	73	65	48	65	99	44	70	99	96	71	50	95

the incorrect detections. A high correct detections score achieved by the SR module proves that this hypothesis requires further evaluation.

Similar potential of the Simulated Recurrence method can be observed when analyzing the results obtained with the Kolmogorov-Smirnov test. Although this univariate test makes many false-positive detections, the Simulated Recurrence module makes significantly less mistakes for most of the scenarios, obtaining a descent sensitivity score.

Lastly, in each scenario at least one method achieves a perfect score. This suggests the implementation of detector ensembles, which could mutually use the strengths of every detection algorithm and achieve higher scores together.

5 Conclusions

Simulated Recurrence as an enhancement of the Detection module performs very well when it is combined with a reliable and not over-sensitive detection algorithm, such as the CNF test.

Also, the method has a filtering potential, which can be observed in the multivariate Wald-Wolfowitz scenario, where the Detection module performs many false-positive detections, while the Simulated Recurrence module identifies almost none of them. It can be inferred, that in case if the SR module possessed the knowledge about all possible concepts, it could deny the detection signal when the new concept could not be identified.

Summing up, the promising scores suggest, that the method is worth investigating and should be a subject for deeper analysis. In current form, it is a solid base for future research, which we would like to direct in the following areas:

1. Expand the experiments by testing many possible concepts. In our paper we are identifying a single concept, which may seem trivial. In the future, we are planning to test whether the same simulated recurrence methods perform with a similar efficiency in the multi-concept scenario.
2. Deploy simulated recurrence to validate the Detector's decisions or even replace the Detector. We have already performed preliminary experiments of using the simulated recurrence as a detector with the methods described in the article. The detection was performed by comparing the p-values obtained by comparing the distributions of data in windows with the reference and artificial datasets. In a few cases, the Detector based on the simulated recurrence has surpassed the original algorithm and in the best cases it achieved around 25% better results, therefore this area also asks for further analysis and evaluation.
3. Evaluate other methods for simulating recurrence. In this paper we have proposed two methods for simulating recurrence, in the future we plan to investigate more approaches, such as simulating only the most frequent features, limiting the attribute values in the concept and including the relationships between the attributes.

4. Apply detector ensembles. Ensembles have proven to be a reliable method for combining classifiers when coping with concept drift [8] and, as the task of concept drift detection is also a classification problem, we plan to deploy this approach in future research.
5. Use other univariate generalization methods and different concept drift detection algorithms. The univariate statistics can be generalized into the multivariate scenarios in a more sophisticated manner, which requires experimental evaluation. Also, there are many more advanced concept drift detection algorithms in the literature, which can be analyzed in combination with simulated recurrence.

Acknowledgements. This work is supported by The Polish National Science Centre under the grant which is realizing in years 2010-2013.

References

1. Newman, D.J., Asuncion, A.: UCI machine learning repository (2007)
2. Dries, A., Rückert, U.: Adaptive concept drift detection. *Stat. Anal. Data Min.* 2(56), 311–327 (2009)
3. Friedman, J., Rafsky, L.: Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics*, 697–717 (1979)
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. Greiner, R., Grove, A.J., Roth, D.: Learning cost-sensitive active classifiers. *Artif. Intell.* 139(2), 137–174 (2002)
6. Hotelling, H.: The Generalization of Student's Ratio. *Annals of Mathematical Statistics* 2(3), 360–378 (1931)
7. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, pp. 97–106 (2001)
8. 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)
9. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures, 4th edn. Chapman & Hall/CRC (2007)
10. Sobolewski, P., Woźniak, M.: Artificial Recurrence for Classification of Streaming Data with Concept Shift. In: Bouchachia, A. (ed.) ICAIS 2011. LNCS, vol. 6943, pp. 76–87. Springer, Heidelberg (2011)
11. Sobolewski, P., Woźniak, M.: Data with Shifting Concept Classification Using Simulated Recurrence. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ACIIDS 2012, Part I. LNCS, vol. 7196, pp. 403–412. Springer, Heidelberg (2012)
12. Vreeken, J., van Leeuwen, M., Siebes, A.: Characterising the difference. In: Berkhin, P., Caruana, R., Wu, X. (eds.) KDD, pp. 765–774. ACM (2007)
13. Zliobaite, I.: Change with delayed labeling: When is it detectable? In: Proceedings of the 2010 IEEE International Conference on Data Mining Workshops, ICDMW 2010, pp. 843–850. IEEE Computer Society, Washington, DC (2010)

DeltaDens – Incremental Algorithm for On–Line Density–Based Clustering

Radosław Z. Ziemiński

Abstract. Cluster analysis of data delivered in a stream exhibits some unique properties. They make the clustering more difficult than it happens for the static set of data. This paper introduces a new DeltaDens clustering algorithm that can be used for this purpose. It is a density–based algorithm, capable of finding an unbound number of irregular clusters. The algorithm’s per–iteration processing time linearly depends on the size of its internal buffer. The paper describes the algorithm and delivers some experimental results explaining its performance and accuracy.

Keywords: Density–based Clustering, On–line Clustering, Data Streams.

1 Introduction

Cluster analysis of data delivered in a stream is a common task in many technical applications. All devices beginning from a simple microphone or temperature sensor to very sophisticated scanning devices used in biotechnology, robotic, particle physics or space exploration register and deliver information in data streams.

The processing of the data stream characterizes some specific properties. They make it different from the processing of static sets of data. Clustered data are often sequentially ordered according to their timestamps in the stream. They can be delivered at different rates during the processing. Furthermore, they may be incoming uninterruptedly for a long time thus they cannot be easily buffered. Additionally, the distribution and shapes of clusters may slowly or rapidly change in time.

These properties imply requirements for an algorithm suitable for the clustering of data delivered in the stream:

Radosław Z. Ziemiński
Institute of Computing Science, Poznan University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland
e-mail: radoslaw.ziembinski@cs.put.poznan.pl

- The algorithm has to incrementally update its internal model of clusters for incoming data. This procedure allows the algorithm to follow-up changes in the data distribution utilizing a limited pool of resources. It should be assumed that data presentation occurs only once during the processing. The algorithm should be capable to detect and describe an unbound number of irregular and evolving clusters.
- The algorithm should keep a constant response time for the processing of incoming data objects. A low stability about this matter may cause lags and data drops during the stream processing. Moreover, the response time should be possibly short to withstand different rates of the data presentation.
- It has to be sufficiently “agile” to capture and describe short living clusters along to some long living ones. Additionally, it should properly recognize cases where the cluster signature was interrupted or distorted over a period. It also has to remove a cluster from its model if it became unsupported by data. Moreover, it has to be robust on noise in data and properly recognize outliers.

Development of the algorithm matching to these requirements can be a difficult task. Many algorithms applied for the clustering of static sets are not suitable because they cannot update their internal models efficiently. Even if the concept of the sliding window is applied, they rebuild each time the model from scratch.

This paper introduces the DeltaDens algorithm that matches to the most of the requirements. It is the density-based algorithm that uses a concept of the dense neighborhood. It implements a simple internal model of clusters which can be efficiently maintained in an incremental way. It also provides a new method of the model handling capable of on-line updating collection of identified clusters.

This paper begins from a brief overview of related works. Afterward, it delivers a short introduction to the DeltaDens algorithm. It follows a presentation of results from experiments and inferred conclusions.

2 Related Works

This paper describes a study of the density-based clustering algorithm. These algorithms have been primarily developed for the clustering of static sets of data e.g. [4] and [6]. They identify adjacent objects with dense neighborhoods. Main parameters of these algorithms are the neighborhood radius and the density threshold. Differently from other families of clustering algorithms, they can find an unbound number of irregular clusters. Their processing time can be compared to one of data sorting algorithms.

Some density-based clustering algorithms have been altered for the stream processing. They usually work in dual on/off-line modes. The on-line component updates the internal model of clusters in an incremental way. Then the model is used in the off-line mode to produce an output set of clusters. This step is often computationally expensive thus it is executed on demand. An example density-based clustering algorithm with capability of the stream processing has been described in

[2]. It was named DenStream and it uses a concept of spherical microclusters to encapsulate small “dense” regions of space. The algorithm maintains a set of dense microclusters. It swiftly updates them for incoming data objects. They contain information for constructing irregular clusters in the off–line mode. Another proposal, D-Stream algorithm is described in [3]. Authors developed a grid structure to find dense regions of space. The algorithm uses it and performs the bin sorting of data objects in the off–line phase. However, it requires special measures to prevent from explosion of the grid size for high–dimensional problems. A more advanced algorithm is introduced in [7]. It can detect clusters of different densities by automatically adjusting the density threshold value.

Beside density–based clustering algorithms, many other algorithms built on different principles exist. For an example, CluStream introduced in [1] uses the k –means algorithm to create microclusters. Almost parameterless is a following Clus-Tree algorithm described in [5]. It maintains a tree structure describing distribution of objects to provide information for the off–line phase.

The DeltaDens algorithm introduced in this paper has some competitive features:

- Its internal model of clusters relies on samples of data objects organized as collection of sorted lists. A new method of objects replacement has been implemented to maintain a limited size of the cluster representation. It delivers reasonable performance and delimits the number of data objects describing clusters.
- The internal model of clusters is built on sorted lists. They can be merged fast if some clusters became connected by incoming data object. Additionally, it introduces new and efficient method of the cluster model partitioning. It is used if the removed object introduced a fragmentation to the cluster model.
- The algorithm implements a history buffer. Its size delimits memory necessary for the processing and per–iteration response time. It allows to find old or exceeding data objects and remove them from the internal model of clusters.
- A grid–based microcluster representation delivers the accurate representation of irregular clusters. Hence, the off–line phase is relatively simple and has a low computational cost.

3 DeltaDens Algorithm

The DeltaDens algorithm performs the clustering of sorted sequence of objects $o_t(a_1, a_2, \dots, a_d)$, where t is unique timestamp, $a_m, m = 1..d$ are attributes describing object o_t and d is the number of attributes. It requires that all attributes must be equipped with metrics. Metrics determine the membership in the neighborhood for a pair of objects. The algorithm accepts four parameters. A history buffer size α delimits the number of objects describing clusters in the internal model. The buffer should be larger if many clusters are expected in the output set or incoming data objects support clusters unevenly. Its size has impact on the per–iteration response time. Time threshold value β allows to find and remove old objects from the internal model. It should be greater if the rate of the data presentation significantly varies or

lower if rapid changes in the clusters layout are expected. Furthermore, μ parameter is the density threshold. It is a smallest number of objects required to recognize a neighborhood as dense one. Remaining ε parameter determines the neighborhood radius. The pair of attribute values is in the neighborhood if a distance between them is less or an equal to ε . The object is in the neighborhood of another one if it is in the neighborhood for all attributes. In a general case, ε can be replaced by array containing radii associated to attributes. They would define the bounding box of the neighborhood. Properties of μ and ε parameters are like ones introduced in [4] and [6]. For some high-dimensional problems, μ should be set to lower values and ε to greater values to overcome the problem of more sparse data objects distribution. However, a greater value of ε may lower the accuracy of the produced clustering.

The algorithm maintains a sorted map B . It contains an association between cluster identifiers and sets of objects representing them $C_k, k = 1..|B|$ (internal models of clusters). The map B is sorted according to clusters identifiers. Moreover, cluster C_k is implemented as array of sorted lists $C_k[m]$, where $m = 1..d$. All objects added to internal models are also added to the history buffer H . The buffer H is sorted according to timestamps. Hence, the number of internal models is limited $k \leq |H|$.

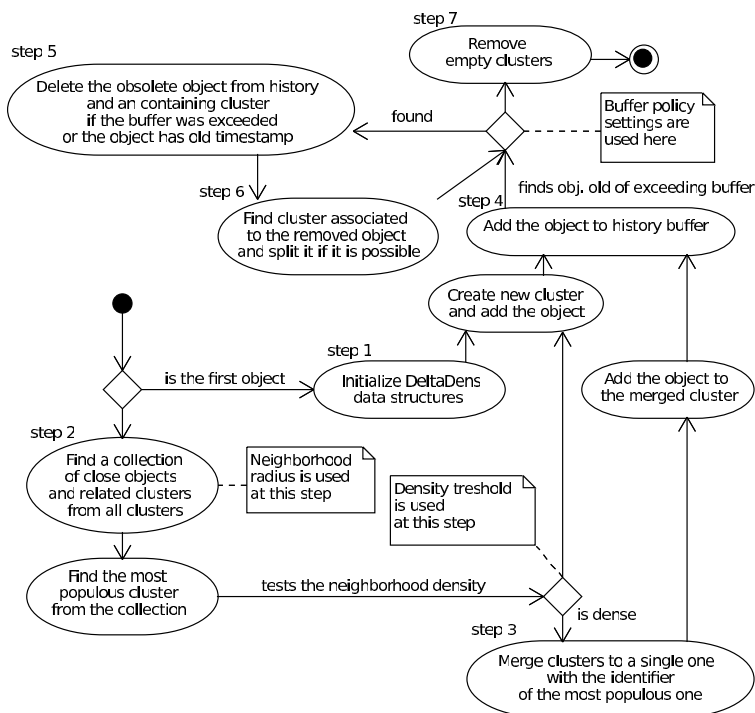


Fig. 1 Iteration of the DeltaDens algorithm

The DeltaDens iteration is illustrated in Fig. 1. Initially, the first internal model of the cluster C_1 is created if the first object o_1 is submitted to the algorithm (step 1). For following objects, the algorithm finds objects that resides in a neighborhood of the incoming object on map B . It sequentially browses B and identifies a neighborhood using μ and ε thresholds for each $C_k \in B$. It can be done fast because all objects are sorted according to particular attribute's values order on each list $C_k[m]$. The objects residing in the neighborhood are found using the binary search algorithm and the sequential neighborhood scanning for each attribute. Then the product is calculated using objects sets intersection for all attributes of C_k . Thereafter, it is integrated for all $C_k \in B$ to find the neighborhood (step 2). If the dense neighborhood was not found on B then it is created a new cluster model containing only the incoming object. Otherwise, all C_k participating to the dense neighborhood are merged. Then the new object is added to the merged cluster (step 3). Because clusters models rely on sorted lists $C_k[m]$ thus their integration can be done fast (merge sort).

However, addition of the incoming object may be changed to the replacement if the neighborhood density exceeded $2 \cdot \mu$. Then the closest object to the incoming one is removed from the merged cluster model. This measure prevents from an overrepresentation of some dense clusters at the cost of more sparse ones. Otherwise, existence of sparse ones could become unnoticeable to the algorithm due to the limited size of H . The merged cluster has an identifier inherited from the most populous cluster C_k contributing to the dense neighborhood. Thus, it is assumed that the identifier of the most populous cluster should be possibly stable. The incoming object is also added to buffer H (step 4).

When the incoming object has been added, others may be removed from the buffer and internal models in the same iteration. Hence, the algorithm sequentially browses buffer H . Old objects o_t are removed if $T - t > \beta$, where T is the current time e.g. the insertion time of the last incoming object. Moreover, if $|H| > \alpha$ then “fifo” rule is applied to remove all objects exceeding capacity of the buffer. Exceeding or old objects are removed from the history H and a corresponding internal cluster model C_k using the binary search algorithm (step 5). Noteworthy, every object o_t from C_k is wrapped in a structure containing information about an identifier of the associated cluster model. This “reverse” reference improves the algorithm's performance. Moreover, the removal of any data object may lead to fragmentation of the cluster model. The following pseudo–code explains how the issue of the fragmentation is handled:

```

procedure removeDataObject( $o_r$ ):
 $C[m][pos]$  - attribute's value of cluster's model list
                for dimension  $m$  and position  $pos$ 
 $M$  - sorted map of key-cluster pairs
 $key$  - cluster identification key
 $d$  - number of attributes (dimensions)

 $M = \{\}$ ;  $split[] = \{\}$ ;  $count = 1$  // splitting information
 $C = B.find(o_r)$  // find related cluster
for  $m = 1..d$  do // process the fragmented source cluster
     $pos = C[m].position(o_r)$  // binary search
    if  $0.25 \cdot |C| < pos < 0.75 \cdot |C|$  then

```

```

     $ln = \max(pos - 1, 1); rn = \min(pos + 1, |C|)$ 
    if  $C[m][rn] - C[m][ln] > \varepsilon$  then // test separation
         $split[m] = pos; count++$ 
C.remove( $o_r$ )
if  $count > 1$  then
    for  $o_t$  in  $|C|$  do
         $key[] = \{\}; count = 1$ 
        for  $m = 1..d$  do
             $pos = C[m].position(o_t)$ 
            if  $pos > split[m]$  then
                 $key[count++] = 0$  else  $key[count++] = 1$ 
         $C_t = M(key)$  // get target cluster or create empty one
            // if it was not on the map
         $C_t.add(o_t)$  // move object to the target cluster
B.remove( $C$ )
B.addAllClusters( $M$ )

```

The algorithm finds if “sparse” pairs of objects adjacent to the removed object are present for every attribute (step 6). If the difference is greater than ε then the cluster model became fragmented. Because the fragmentation can be possible at many dimensions at once thus a *key* array is filled according to the pseudo-code. It describes a relative position of the processed object to the removed one and identifies the target cluster model on the map *M*. Afterward, all objects from the fragmented cluster model are distributed to target ones stored on *M*. Then target clusters are added to the map *B* and the fragmented cluster model is removed. The map *B* is cleaned and empty clusters models are removed if some left after the objects removal cycle (step 7). The iteration ends.

The per-iteration pessimistic cost is $O(|H| \cdot d)$ (the neighborhood extraction cost for large ε values). The operation cost on the sorted map or list is $O(\log(l) \cdot d)$, where $l \leq |H|$ is a size of the list. The bound on memory is $O(|H| \cdot d)$ (sorted collections are implemented as sorted arrays of objects).

Clusters in the DeltaDens algorithm are described by sets of objects $C_k, k = 1 \dots |B|$ obtained from the on-line phase. However, a specific simplification of the clustering result is required in many applications. Hyperspherical clusters representation as proposed here is used in the MOA framework. It is expected by the framework to compare DeltaDens to other algorithms.

The off-line postprocessing of the clustering result is different for hyperspherical clusters and sets of microclusters. In the first case it produces a result set *R* of hyperspheres H_k for all $C_k : |C_k| > \mu$. This threshold prevents from passing a noise to the result. The center of H_k is calculated as the mean point of the C_k set. Then the radius of H_k is calculated according to formula $2 \cdot RMSE + \varepsilon$, where *RMSE* is the root mean square error calculated from the cluster center. Moreover, if the one small cluster overlaps significantly with some larger cluster then it is not added to the output to cut the redundancy. Optionally, center and radius calculations may include objects timestamps as weights. Then older objects would participate less to the hypersphere position and size.

A more complex process delivers sets of hyperspherical microclusters. The post-processing procedure begins from filtering out the noise like in the earlier case. Then

the space occupied by data objects is partitioned on equal hypercubes. Their side length is $4 \cdot \varepsilon$. Coordinates of cubes build a key for a map storing counters (it reduces memory consumption). If data object from C_k resides in the particular hypercube then the related counter is incremented. This process is fast because it uses the bin sorting. If the algorithm dispatches all data objects from C_k then it creates a microcluster from every non–empty hypercube. The center of the microcluster is the same as the center of the hypercube and the radius is $(2 + d/10) \cdot \varepsilon$. Experimental choice of the radius and the hypercube cell length was for the DeltaDens evaluation in MOA. The obtained set of microclusters describes irregular shape of the cluster C_k . Noteworthy, $|H|$ delimits the largest number of microclusters.

4 Experimental Evaluation

The first group of conducted experiments evaluated the computational efficiency of DeltaDens. Hence, the execution time was measured for different data streams. The data streams were generated from “moving” hyperspherical clusters within hypercube of length 1.0. The margin preserved for each side of hypercube was equal to 0.2. The clusters dispersion was Gaussian with radius equal to 0.03. The time horizon of the generated stream was dependent on the stream size.

DeltaDens settings were the same in all experiments measuring the performance $\varepsilon = 0.02$, $\mu = 2$, $\alpha = 1000$ and $\beta = 1000$. It helped to correlate the results. All obtained results are presented in Fig. 2. Every result is an average from 8 measurements done on random streams generated for the same configuration. Three quantities were measured. The feeding time describes a one required for the updating of internal models for incoming objects. The model preparation time reflects one taken by preparation of a hyperspherical clusters model from the algorithm’s internal models. Finally, the microclusters preparation time describes one required for preparation of microclusters sets.

In the first experiment data objects had 3 attributes. The stream contained 50000 objects but the number of clusters was gradually increasing during the experiment. Top–left chart of Fig. 2 presents results. It can be observed that the number of clusters has a low impact on the per–iteration execution time. It is increasing linearly if the number of clusters is growing. An underlying issue affecting the performance is the larger number of internal clusters models required to describe more complex data distribution. Denser clusters can explain higher cost for a low number of clusters (the presentation rate was the same in all cases). A small set of denser clusters is handled longer because the cost of the lists sorting in internal models of clusters is higher.

The second experiment was a kind of the endurance test. A long stream of objects was divided into packets containing 1000 objects each and measurements were done for each packet. Top–right chart of Fig. 2 delivers results. They show that the algorithm maintains the stable per–iteration execution time during the long processing. It depends mainly on $|H|$.

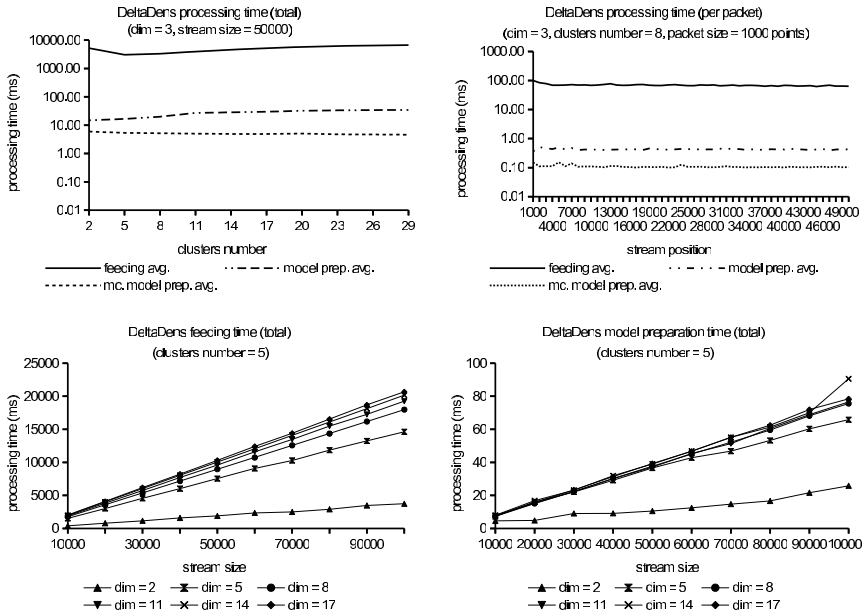


Fig. 2 Measurements of the DeltaDens processing efficiency

The third experiment measured the total time of the stream processing for data of different sizes and dimensionality. Bottom charts of Fig. 2 present results. According to results, the total execution time is growing linearly to the stream size. It is not surprising in the light of previous results. However, it is also decreasing if dimensionality of the data set is growing. A more sparse environment created at high-dimensional problems can explain this result. In such cases, it is less likely to find the dense neighborhood for the same parameters. Thus, the algorithm performs relatively fast. However, implementation of the algorithm includes d sorted lists for each internal cluster model. Therefore, the cost could grow linearly to the number of dimensions only if density of clusters in the stream would be compensated according to the problem dimensionality.

Experiments evaluating DeltaDens results were conducted using the open source MOA framework. It was obtained from [8]. Its application required implementation of the MOA interface for DeltaDens. The framework allows to compare results produced by DeltaDens to DenStream, CluStream and ClusTree (implemented in MOA). The experiments used some evaluation measures implemented in MOA. The Redundancy measure indicates if objects were assigned to more than one cluster at once. The GPrecision measure describes proportion of non-noise objects number covered by the clustering to the number of all objects covered by the clustering. The GRecall considers in the denominator the number of all non-noise objects covered by the clustering. F1-P, F1-R and Purity measures are calculated using the cluster membership matrix (CMM). The Purity is accumulation of *precision* for the CMM. The F1-P is accumulation of F_1 measure values calculated

Table 1 Evaluation of clustering results and the processing times (include evaluations)

Dimensions	F1-P		F1-R		Purity		GPrecision		GRecall		Renundancy		Proc. time (sec)	
	3	8	3	8	3	8	3	8	3	8	3	8	3	8
Hyperspherical representation (5 clusters embedded)														
DeltaDens	0.95	1.00	0.87	1.00	0.95	1.00	1.00	1.00	0.91	1.00	0.00	0.00	82.9	149.0
CluStream	0.92	0.99	0.87	0.97	0.91	1.00	0.98	1.00	0.93	0.97	0.01	0.00	65.4	151.0
DenStream	0.69	0.64	0.69	0.60	0.99	1.00	1.00	1.00	0.54	0.45	0.00	0.00	299.5	557.3
ClusTree	0.90	0.99	0.85	0.91	0.89	1.00	0.98	1.00	0.93	0.91	0.01	0.00	21.5	30.5
Hyperspherical representation (14 clusters embedded)														
DeltaDens	0.91	1.00	0.75	1.00	0.94	1.00	0.99	1.00	0.82	1.00	0.00	0.00	88.9	103.1
CluStream	0.89	0.98	0.79	0.94	0.90	1.00	0.97	1.00	0.84	0.93	0.01	0.00	63.5	169.8
DenStream	0.56	0.54	0.55	0.50	0.99	1.00	1.00	1.00	0.42	0.37	0.00	0.00	223.7	436.4
ClusTree	0.87	0.97	0.79	0.88	0.88	1.00	0.98	1.00	0.85	0.87	0.01	0.00	33.4	44.7
Microcluster representation (5 clusters embedded)														
DeltaDens	0.21	0.32	0.67	0.72	0.98	1.00	1.00	1.00	0.99	0.95	0.17	0.17	83.0	149.2
CluStream	0.17	0.12	0.59	0.37	0.99	1.00	0.92	0.81	0.62	0.26	0.03	0.00	75.4	166.0
DenStream	0.07	0.09	0.25	0.26	1.00	1.00	1.00	1.00	0.61	0.57	0.34	0.33	652.7	1307.6
ClusTree	0.40	0.61	0.58	0.73	0.99	1.00	0.92	0.93	0.74	0.76	0.26	0.16	48.7	62.2

as $F_1 = (2 \cdot \text{precision} \cdot \text{recall}) / (\text{precision} + \text{recall})$ for the CMM. The F1-R accumulates greatest F_1 measure values calculated for the generated clusters classes. The MOA source code has the complete definition of measures.

Two kinds of experiments were conducted. They evaluated different representations of the clustering results: hyperspherical clusters and sets of microclusters. Table 1 delivers results from these experiments. The MOA framework provided all measurements. Experiments relied on artificially generated data where the stream contained 500000 objects and $d = \{3, 8\}$. The data generator implementation was the class RandomRBFGeneratorEvents. The settings for the algorithm were almost default with following exceptions: $\text{numCluster} = \{3, 14\}$, $\text{modelRandomSeed} = 8$ and $\text{instanceRandomSeed} = 6$. Configuration of algorithms was similar to MOA default. DeltaDens: $\varepsilon = 0.02$, $\mu = 2$, $\alpha = 1000$ and $\beta = 1000$. DenStream: $\text{horizon} = 1000$, $\varepsilon = 0.02$, $\text{minPoints} = 10$, $\beta = 0.001$, $\mu = 1$ and $\text{initPoints} = 1000$. Cases containing more than 14 clusters could not be evaluated with DenStream because MOA refused to give mean statistics of the processing. CluStream: $\text{horizon} = 1000$, $\text{maxNumKernels} = 100$ and $\text{kernelRadiFactor} = 2$. ClusTree: $\text{horizon} = 1000$ and $\text{maxHeight} = 8$. Noteworthy, a higher number of clusters and a lower number of dimensions have negative impact on clusters separability.

According to obtained results, the DeltaDens algorithm often outperformed other algorithms for problems of different dimensionality and for different numbers of clusters embedded in the stream. In many cases, the regular grid of microclusters used in DeltaDens was more accurate than heuristics used in other algorithms. Furthermore, a compression algorithm for adjacent microclusters may improve the microclusters generation process. It should cut their number and keep the accuracy of the cluster description.

OpenJDK 6 with the default configuration of the garbage collector was the implementation and execution environment for all presented experiments.

5 Conclusions

The DeltaDens algorithm straightforwardly applies to the clustering of data delivered in the stream where numeric attributes describe processed objects. However, introduction of similarity thresholds specified independently for each attribute alone (ϵ radii) extends it easily to other types of information. The algorithm modifies its internal clusters models iteratively for incoming data objects. It recognizes objects timestamps.

Conducted experiments confirmed that per-iteration execution costs met assumed requirements for this type of algorithms. DeltaDens response time is about constant during the stream processing for equally sized packets of incoming objects. The processing cost and memory consumption depend linearly on the product of the internal buffer size $|H|$ and the problem dimensionality. Moreover, evaluation of the clustering results confirmed that DeltaDens can deliver comparable or even better results than ClusTree, CluStream and DenStream. The proposed grid-based microcluster structure is accurate and fast to produce. Although the number of microclusters can be large for the high-dimensional problem, it is limited by $|H|$.

References

1. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proc. of the 29th Int. Conf. on Very Large Data Bases, vol. 29, pp. 81–92. VLDB Endowment (2003)
2. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-Based Clustering over an Evolving Data Stream with Noise. In: Proc. of the Sixth SIAM Int. Conf. on Data Mining, pp. 328–339. SIAM (2006)
3. Chen, Y., Tu, L.: Density-based clustering for real-time stream data. In: Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142. ACM (2007)
4. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press (1996)
5. Kranen, P., Assent, I., Baldauf, C., Seidl, T.: The ClusTree: indexing micro-clusters for anytime stream mining. *Knowledge and Information Systems* 29, 249–272 (2011)
6. Sander, J., Ester, M., Kriegel, H.-P., Xu, X.: Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Mining and Knowledge Discovery* 2, 169–194 (1998)
7. Wan, L., Ng, W.K., Dang, X.H., Yu, P.S., Zhang, K.: Density-based clustering of data streams at multiple resolutions. *ACM Trans. Knowl. Discov. Data* 3, 14:1–14:28 (2009)
8. MOA (Massive Online Analysis), software release (March 2012), <http://moa.cs.waikato.ac.nz/downloads/>

Part III
Ontologies Meet Advanced Information
Systems

Introducing Artificial Neural Network in Ontologies Alignment Process

Warith Eddine Djeddi and Mohamed Tarek Khadir

Abstract. During automated/semi-automated alignment across myriad ontologies, different similarity measures of different categories such as string, linguistic, and structural based similarity measures, contribute each to some extent to alignment results. A weights vector must, therefore, be assigned to these similarity measures, if a more accurate and meaningful alignment result is favored. It is not trivial to determine what those weights should be, and current methodologies depend a lot on human heuristics and/or prior domain knowledge. In this paper, we take an artificial neural network approach to learn and adjust these weights, with the purpose of avoiding some disadvantages in both rule-based and learning-based aligning algorithms. XMap++ is applied to benchmark tests at OAEI campaign 2010. Results show that neural network boosts the performance in most cases, and that the proposed novel approach is competitive with top-ranked system.

1 Introduction

Semantic web researchers are proposing more meaningful service descriptions, by annotating them with a semantic description of their functionality. The meaning of these semantic descriptions is, however, grounded on the availability of domain ontologies as semantic interoperability in the World Wide Web (WWW) is still limited due to the heterogeneity of information. Ontology, a formal, explicit specification of a shared conceptualization [1], has been suggested as a way to solve the problem. For this reason, before being able to combine similar ontologies, a semantic and structural mapping between them has to be established. The process of establishing such a mapping is called ontology alignment. It will become increasingly significant as the semantic web evolves, it is already an active research area and several

Warith Eddine Djeddi · Mohamed Tarek Khadir
LabGED, Computer Science Department, University Badji Mokhtar, Po-Box 12,
23000 Annaba, Algeria
e-mail: {djeddi, khadir}@labged.net

automatic or semi-automatic ontology alignment tools have been proposed. Most of the tools rely on heuristics that detect some sort of similarity in the description of the concepts and the structure of the ontology graphs, by using e.g. string and graph matching techniques. Comprehensive surveys of the state of the art ontology mapping approaches can be found in [2]. Though the state of the art approaches have made significant progresses in ontology mapping, they suffer from many limitations. (1) Ontology mapping approaches that use multiple mapping strategies meet the problem of aggregating multiple similarities, with manually setting parameters which is impractical due to its inability to adapting to different ontology mapping tasks. (2) The problem of such approaches is the difficulty of collecting sufficient training data that may it-self incur a substantial effort. (3) A further problem is that even within a domain the successful configurations for one match problem do not guarantee sufficient match quality for different problems, especially for matching large schemas. Therefore, one would need methods to preselect suitable and sufficient training correspondences for a given match task, which is an open challenge. To overcome the limitations, in this paper we propose a new automatic techniques to train and generates the string, linguistic and structural weights. Furthermore, the burden of manual selection of weights has been definitely eliminated. In this paper, we have combined different weights of string-based, linguistic and structural categories into one input sample. The ensemble method is an active research area which gives better performance than a single classifier [3]. Some research works have shown that using a single classifier performing well may not be the optimal choice [4]. Therefore, the main contributions of this work are: (1) Our approach integrates an Artificial Neural Network (ANN) technique in our algorithm, such that the weights mentioned above can be learned instead of being specified by a human in advance. (2) Moreover, our learning technique is carried out based on the ontology schema information alone, which distinguishes it from most other learning-based algorithms [5], [6] which rely in most cases on ontology instances. In order to avoid the problem of missing instances data (either in quality or in quantity), which is common for real-world ontologies, our weight learning technique is carried out at the schema level instead of the instance level. (3) The additional possibility to obtain goal-driven results, thus optimize some of the characteristics of an output alignment. (4) We provide results following a standard benchmark to enable the comparison with other approaches. The rest of this paper is organized as follows. Section 2 describes the related works for ontology alignment. Section 3 gives an overview of our approach, and in section 4 discusses the strategy in applying machine learning techniques in our algorithm. Section 5 reports the experiments conducted and section 6 analyzes the results. Finally section 7 concludes on the results.

2 Related Work

Ontology matching [2] is used for creating mappings between ontologies, where ontologies alignment enables the knowledge and data expressed in the matched

ontologies to be interoperated. A major insight of Ontology Alignment Evaluation Initiative (OAEI) [2], concludes that there is no best method or system for all existing matching problems. The factors influencing the quality of alignments range from differences in lexical similarity measures to variations in alignment extraction approaches. During the past years, a lot of research has been devoted to developing highly sophisticated tools for performing ontology matching automatically [7]. Those tools are able to produce high-quality mappings between ontologies, given that their parameters (such as weights and thresholds used to compute the mappings) are tuned well. Such a tuning, however, is often complicated, since it involves the setting of many parameters and requires a lot of detail knowledge about the underlying algorithms and implementations. Bellahcene and Duchateau in [8] provides an over-view of recent approaches including tuning frameworks. Most previous approaches for automatic tuning apply supervised machine learning methods. They use previously solved match tasks as training to find effective choices for matcher selection and parameter settings such as similarity thresholds and weights to aggregate similarity values, e.g. [9]. The PRIOR+ system [6] propose a new weight assignment method to adaptively aggregate different similarities and then adopt a neural network based constraint satisfaction model to improve overall mapping performance from aggregated results. In [5] an automatic ontology alignment method based on the recursive neural network model that uses ontology instances to learn similarities between ontology concepts is proposed. RiMOM [10] using Bayesian decision theory in order to generate an alignment between ontologies, and additionally accepts user input to improve the mappings. GLUE [11] employs machine learning techniques that analyze the taxonomy and the information within concept instances of ontologies. The problem is that those kinds of proposals use weights is initially carried out by a human, while using our approach involves to compute the weights in an automatic way, so the process can be more flexible, at least, in real scenarios.

3 Overview of Our Approach

3.1 Details of XMap++

XMap++ (eXtensible Mapping) is a system for ontology alignment that performs semantic similarity computations among terms of two given ontologies. In XMap++'s previous version [12], we propose a strategy selection method to automatically combine the matching strategies based on the weight of the linguistic affinity W_{LA} . This weight is calculated as given by equation 1.

$$W_{LA} = \frac{\textit{linguistic similarity measure}}{\textit{maximum(linguistic similarity measure, structural similarity measure)}} \quad (1)$$

In its current version, we integrate machine learning techniques, such that the weights W_{LA} can be learned from training examples, instead of being calculated as in equation (1). We build a 3-dimension vector for each concept, and each dimension records one semantic aspect, which represent a combination of different categories of similarity measures, such as string, linguistic and structural methods.

3.2 Feature Selection

String-based, linguistic and structural methods are three different categories of measuring similarities in ontology alignment. Each method returns a similarity value in the range of [0, 1] for a given entity pair from two ontologies. These methods are briefly introduced in the following subsections.

3.2.1 Feature Selection

The string similarity methods compare the concepts textual descriptions associated with the nodes (labels, names, identity, etc) of each ontology.

3.2.2 Linguistic Similarity

WordNet [13] is currently the most popular semantic resource in the computational linguistics community. A known problem of WordNet is that it is too finegrained in its sense definitions (many classes in WordNet are very generic). For instance, it does not distinguish between homographs (words that have the same spelling and different meanings) and polysemes (words that have related meanings) [14]. Often the same word placed in different textual contexts assumes completely different meanings. In order to deal with lexical ambiguity, this approach introduces the notion of "scope" of a concept which represents the context where the concept is placed. In our approach, the similarity between two entities of different ontologies is evaluated not only by investigating the semantics of the entities names, but also taking into account the local context, through which the effective meaning is described. The context is the set of information (partly) characterizing the situation of some entity [15]. The notion of context is not universal but relative to some situation, task or application [16], [17]. In particular, the neighborhood of a term (immediate parent and children in the "is a" hierarchy) may be especially important.

Figure 1 sketches the idea. The scope defines a round area composed of all concepts that are connected directly or indirectly to the central node. This area represents the context. Increasing the radius means to enlarge the scope (i.e. this area) and, consequently, the set of neighbour concepts that intervene in the description of the context. The value of linguistic methods is added to the linguistic matcher or structure matcher in order to enhance the semantic ambiguity during the comparison process of entities names.

3.2.3 Structural Similarity

Ontology alignment solely based on string and linguistic similarities may provide incorrect match candidates. Structural matching is used to correct such match candidates based on their structural context. The structural approach matches the nodes based on their adjacency relationships. The relationships (e.g., subClassOf and is-a) that are frequently used in the ontology serve as the foundation of structural matching. XMap++ algorithm values the semantic relation between two concepts while taking in consideration the types of cardinality constraints and values between their properties [12].

4 Machine Learning Method

Here, supervised machine learning methods are utilized to extract the optimal model of compound metrics. This means that the training algorithm is given a training set of inputs and the ideal output for each input. The feedforward neural network, or perceptron, is a type of neural network first described by Warren McCulloch and Walter Pitts in the 1940s. The feedforward neural network, and its variants, is the most widely used form of neural network. It is often trained with the back propagation training technique, given the celebrated Multi-Layered Perceptron (MLP). More advanced training techniques may be used, such as resilient propagation. The Resilient Propagation Training (RPROP) [18] algorithm is usually the most efficient training algorithm provided by the framework Encog [19] for supervised feedforward neural networks. One particular advantage to the RPROP algorithm is that it requires no setting of parameters before using it. There are no learning rates, momentum values or update constants that need to be determined. This is good because it can be difficult to determine the exact learning rate that might be optimal.

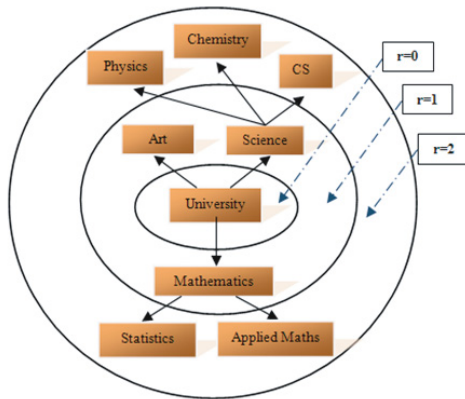


Fig. 1 A sketched ontology and the scope of concept University at different radius r

4.1 Concept Similarity Matrix

To construct the similarity matrix, similarity measures (section 1.3.2) are applied to a pair of ontologies selected from the data sets. Similarity matrix is a table with m rows and n columns, where m is the number of given entity pairs and n is the number of applied features (similarity measures). Having provided the similarity matrix and target values, the problem would be reduced to a supervised learning task comprised of training and testing phases.

Definition 1. Let α and β respectively concepts of the ontology O_1 and O_2 . Let s_1 , s_2 and s_3 respectively the metrics of the string, linguistic and structural similarity. Let w_1 , w_2 and w_3 respectively the weights of the string, linguistic and structural similarity. After s_1 , s_2 and s_3 between two concepts, α and β , are calculated, the similarity value s is obtained as the weighted sum of s_1 , s_2 and s_3 :

$$s = \sum_{i=1}^3 (w_i s_i) \quad (2)$$

Where $\sum_{i=1}^3 w_i = 1$. Notice that w_i are randomly initialized and will be adjusted through a learning process (see section 4.2 below). For two ontologies being matched, O_1 and O_2 , we calculate the similarity values for pairwise concepts. Then we build a $n_1 \times n_2$ matrix M to record all values calculated, where n_i is the number of concepts in O_i .

4.2 MLP Network Design

We regard the hypothesis space in this learning problem as a 3-dimensional space consisting of w_1 , w_2 , and w_3 , i.e., a set of weight vectors. Our objective is to find the weights that best fit the training examples. The used neural network corresponds to a MLP, which consists of multiple layers of nodes in a directed graph, each layer fully connected to the next one. Each connection (synapse) has an associated weight. In our particular situation, the network is composed of three layers: input, hidden, and output layer (with three, four, and one neurons respectively; additionally two bias neurons are used in the input and hidden layer respectively). Sigmoid activation function scales the output from one layer before it reaches the next layer. We adopt a three-layer 3×1 network in XMap++, as shown in Fig. 2

The input into this network is a vector, which consists of s_1 , s_2 , and s_3 , representing the similarity in string, linguistic, and properties, respectively, for a given pair of concepts. The output from this network is s , the similarity value between these two concepts as given by equation (2). To train the neural network, we must construct an object that contains the inputs and the expected outputs. To construct this object, we must create two arrays. The first array will hold the input values for the our neural network. The second array will hold the ideal outputs for each corresponding

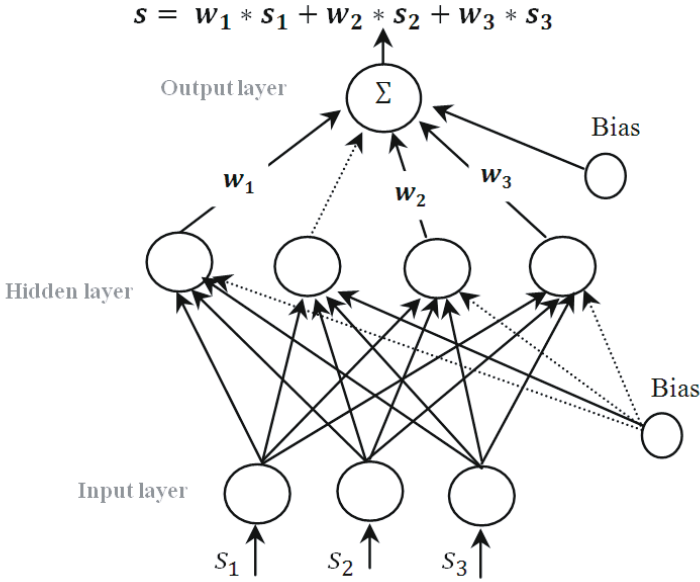


Fig. 2 Neural network structure

input values. These will correspond to the possible values for XMap++. Now that the training set has been created, the neural network can be trained. Training will continue for many iterations, or epochs, until the error rate of the network is below an acceptable level (i.e error = 0.1). For this example we are going to use Resilient Propagation (RPROP). Once the neural network has been trained, it is ready for use.

5 Experiments and Evaluation Criteria

5.1 Implementation and Setting

XMap++ is implemented as a Protege (<http://protege.stanford.edu/>) PlugIn and is tested under Protege 3.4. Moreover, the parameters taken by the approach (i.e. weights, thresholds and the value of radius) were tuned and set depending on the type of information contained in the ontologies to be mapped. The thesaurus WordNet (<http://wordnet.princeton.edu/>) (version 2.1) is optionally used to calculate the lexical similarities between each pair of concepts and properties, in order to derive semantic similarity measures. Finally to create and manipulate neural networks we use the framework Encog (<http://www.heatonresearch.com/encog>).

5.2 Data Sets and Evaluation Criteria

To evaluate our approach we have applied the benchmark tests from OAEI ontology matching campaign 2010. This test set consists of one reference ontology o_R (33 classes, 59 properties, 56 individuals and 20 anonymous individuals), for a bibliographic domain, to be compared with other test ontologies o_T . Some introduced changes include, for example, the extension, or shrinkage of the ontology hierarchy, the use of synonyms, foreign names, removal of class properties and many more. The benchmark tests can be divided into 5 groups as shown in Table 1.

Table 1 Overview of OAEI benchmark tests

Tests	Description
# 101-104	o_R and o_T have the same representation and conceptualisation
# 201-210	o_R and o_T have the same structure but different linguistic
# 221-247	o_R and o_T have the same linguistics but different structure
# 248-266	Both structure and linguistics are different between o_R and o_T
# 301-304	o_T are real-life bibliographic ontologies

We adopt the evaluation criteria used by the campaign. That is, standard evaluation measures precision, recall and f-measure will be computed against the reference alignments.

6 Experimental Design and Results

Here, two experiments have been conducted. First experiment addresses an aspect which has its impact on the training model and the second focuses on testing the training dataset. A. First experiment: The first experiment has simply chosen the optimum model based on string, linguistic and structural similarity measures, which are mentioned in section (3.2). The value of the weights w_1 , w_2 , w_3 and the learning rate η are initialised randomly by the RPROP method. The obtained similarity matrix then aggregated via classification. After adjusting classifiers' parameters, training model was obtained. B. Second experiment: This experiment explores the effect of training samples quantity on the quality of final trained model. In this experiment, the number of entity pairs is increased by using other ontologies such as tests # 102 and #103, i.e. entity pairs extracted from (#101, #102) and (#101, #103). So the diversity of instances in training phase is widened. To avoid training the model with similar input samples, those samples from #102 and #103 ontologies which represent the highest variances are selected.

6.1 The Improvement of Artificial Neural Network in XMap++

As is shown on Fig. 3 tests #103 and #104, reach a full recall, because XMap++ is tested with simple and similar names. For #201 and #202, XMap++ is not good at precision and recall, both down to (0.11) and (0.12) for XMap++ (with Artificial Neural Networks termed from now on N-XMap++), because the names of classes/properties have been "removed".

The #204 test is a naming conventions test, and the result is encouraging for XMap++ (or N-XMap++); the proposed technique gives good precision, the low recall is due to using some short cuts which is not included in the dictionary such as 'MScthesis' with 'MasterThesis'. As is shown on Fig. 3, tests #103 and #104, reach a full recall, because XMap++ is tested with simple and similar names. For #201 and #202, XMap++ is not good at precision and recall, both down to (0.11) and (0.12) for XMap++ (with Artificial Neural Networks termed from now on N-XMap++), because the names of classes/properties have been "removed". The #204 is a naming conventions test, and the result of this test is encouraging for XMap++ (or N-XMap++); the proposed technique gives good precision, the low recall is due to using some short cuts which is not included in the dictionary such as 'MScthesis' with 'MasterThesis'. Test #205 is a synonyms test, the two implemented systems shows good re-sults based on the retrieved synonyms from WordNet, in which, for example, 'frequency' and 'periodicity' are matched. In fact, concerning the test case #205, the high scale of the recall is explained through the searching for WordNet synonyms based not only on the full labels of entities but also based on the context were the entities are placed which reduce the incorrect correspondences generated by the system. Tests (#221, #222, #223, #224, #225 and #228) achieve full recall and precision. These tests use same string properties with different structure representation and eliminations. #230 is an expansion of classes' components and strings properties test, and the result gives good precision and good recall. The proposed technique resolve some matching difficulties such as matching 'Institution' with 'institutionName', 'Organization' with 'organizationName' and 'Journal' with

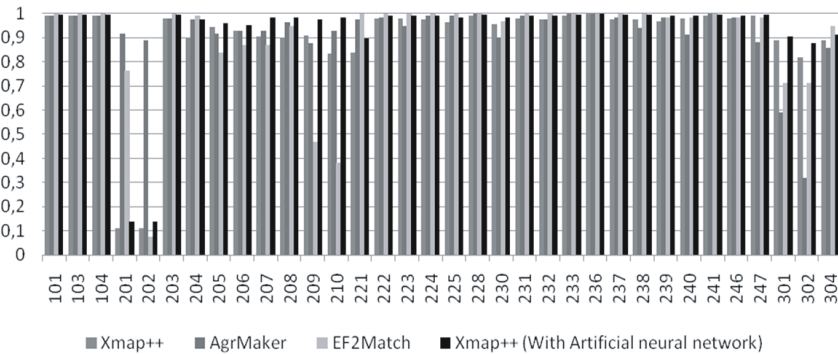


Fig. 3 The comparison of the F-measure of 4 ontology alignment tools on each OAEI benchmark test

'JournalName'. Series (#301-304) represent real-life ontologies modeled by different institutions but for the same domain of bibliographic metadata. In these tests, precision ranges from (0.85) to (0.9) and recall stays between (0.75) and (0.9). XMap++ just can find equivalent alignment relations. However, the inclusion alignments cannot be generated. For #301 and #302, XMap++ finds most correct alignments, but it also returns some wrong results. The alignment results for #303 are far from satisfactory. We think the reason might be that the test #303 is no individuals and with shallow class hierarchy, and there are no direct connections between the classes and properties. Moreover, it is clear that the two systems are efficiently processing tests #301, #302 and #304. Thus, this proves that the semantically context based-approach developed in XMap++ is appropriate for real alignment. As observed from the Table 2, the proposed system has stability characteristics with the different types of tests (except those have no labels).

Table 2 The improvement induced by Neural Networks on XMap++ results

Alignment Tools	Precision	Recall	F-Measure
XMap++	0,9	0.89	0.89
XMap++ (With ANN)	0.93	0.92	0.92
AgrMaker	0.96	0.91	0.92
EF2Match	0.98	0.86	0.89

As it is shown in Table 2, a direct comparison between the XMap++ without ANN, and XMap++ with ANN, shows that the addition of ANN does not has a negative effect on the algorithm but, on the contrary, leads to slightly better results. Such results indicate also that the new approach leads to a better recall, at the cost of precision. If we calculate percentage improvement, that is 3%, 3% and 3% for precision, recall, and f-measure, respectively.

6.2 The Comparison between the N-XMap++ and Top Ranked Systems in OAEI Campaign 2010

We have chosen the alignments generated by the four best matchers that have participated in the 2010 OAEI conference track [7]: AMaker, Asmov and Eff2Match. The results in Table 2 shows the overall f-measure of XMap++ with ANN (0.92) is competitive to that of AgrMaker (0.92) and EF2Match (0.89). Whereas, the precision and the recall of AgrMaker and EF2Match are slightly superior to our algorithm, because they perform perfectly on tests #201 and #202. Comparing N-XMap++, EF2Match and AgrMaker in terms of performance, as it is shown in Fig. 3 and for the tests #203 to #301, we found that our given results are most of the time superior or equals to EF2Match and AgrMaker results.

7 Conclusions and Future Work

In this paper we proposed a novel neural network based approach to search for a global optimal solution that can satisfy ontology constraints as many as possible. We exploit our approach to learning the weights for different semantic aspects of ontologies, through applying an artificial neural network technique during the ontology alignment. This, in turn, increases the discrimination ability of the model and enhances the system's overall accuracy. Therefore we tackle the difficult problem of carrying out machine learning without help from instance data. Also skip over the use of human heuristics and/or domain knowledge to predefine the weights of different categories. We explain and analyze our algorithm in detail, and our experimental results on OAEI (2010) benchmark tests show the approach is promising and constitute our starting point for future explorations in the use of neural networks for computing similarities. XMap++ adopts vectors to record semantic aspects, it is therefore not difficult to handle if more relationships are to be taken into consideration. What needs to be done is for us to expand the current vectors into more dimensions to hold more semantic aspects. Nevertheless, an ANN with multiple layers might be necessary in this case.

References

1. Gruber, T.: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5(2), 199–220 (1993)
2. Euzenat, J., Shvaiko, P.: *Ontology Matching*. Springer, Heidelberg (2007)
3. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20, 226–239 (1998)
4. Tumer, K., Ghosh, J.: Classifier Combining: Analytical Results and Implications. In: 13th National Conference on Artificial Intelligence, Working Notes from the Workshop, Integrating Multiple Learned Models, Protland, Oregon (1996)
5. Chortaras, A., Stamou, G., Stafylopatis, A.: Learning Ontology Alignments Using Recursive Neural Networks. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) *ICANN 2005, Part II. LNCS*, vol. 3697, pp. 811–816. Springer, Heidelberg (2005)
6. Mao, M., Peng, Y., Spring, M.: An Adaptive Ontology Mapping Approach with Neural Network based Constraint Satisfaction. *Journal of Web Semantics* 8(1), 14–25 (2010)
7. Euzenat, J., Ferrara, A., Meilicke, C., Pane, J., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., Trojahn, C.: Results of the Ontology Alignment Evaluation Initiative 2010. In: *Proceedings of the Fifth International Workshop on Ontology Matching, OM 2010. CEUR-WS*, vol. 689 (2010)
8. Bellahsene, Z., Duchateau, F.: Tuning for schema matching. In: Bellahsene, Z., Bonifati, A., Rahm, E. (eds.) *Schema Matching and Mapping. Springer Data-Centric Systems and Applications Series* (2011)
9. Duchateau, F., Coletta, R., Bellahsene, Z., Miller, R.J.: (Not) yet another matcher. In: *Proc. CIKM*, poster paper (2009)
10. Li, Y., Li, J.Z., Zhang, D., Tang, J.: Result of Ontology Alignment with RiMOM at OAEI'06. *Ontology Matching* (2006)
11. Doan, A., Madhavan, J., et al.: Learning to match ontologies on the semantic web. *VLDB Journal* 12(4), 303–319 (2003)

12. Djeddi, W., Khadir, M.T.: A Dynamic Multistrategy Ontology Alignment Framework Based on Semantic Relationships using WordNet. In: Proceedings of the 3rd International Conference on Computer Science and its Applications, CIIA 2011, Saida, Algeria, December 13-15, pp. 149–154 (2011)
13. Fellbaum, C.: WordNet: An electronic lexical database. MIT Press, Cambridge (1998)
14. Jiamjitvanich, K., Yatskevich, M.: Reducing polysemy in WordNet. In: Proceedings of the 4th International Workshop on Ontology Matching, OM 2009, Washington DC, USA, pp. 260–261 (2009)
15. Dey, A., Salber, D., Abowd, G.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction* 16, 97–166 (2001)
16. Dourish, P.: Seeking a foundation for context-aware computing. *Human-Computer Interaction* 16(2-3) (2001)
17. Chalmers, M.: A Historical View of Context. *Computer Supported Cooperative Work* 13(3), 223–247 (2004)
18. Reidmiller, M., et al.: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm. In: *IEEE Inter. Conf. on Neural Network*, pp. 586–591 (1993)
19. Heaton, J.: *Programming Neural Networks with Encog3 in Java*, 2nd edn. (2011)

Time Integration in Semantic Trajectories Using an Ontological Modelling Approach

A Case Study with Experiments, Optimization and Evaluation of an Integration Approach

Rouaa Wannous, Jamal Malki, Alain Bouju, and Cécile Vincent

Abstract. Nowadays, with a growing use of location-aware, wirelessly connected, mobile devices, we can easily capture trajectories of mobile objects. To exploit these raw trajectories, we need to enhance them with semantic information. Several research fields are currently focusing on semantic trajectories to support queries and inferences to help users for validating and discovering more knowledge about mobile objects. The inference mechanism is needed for queries on semantic trajectories connected to other sources of information. Time and space knowledge are fundamental sources of information used by the inference operation on semantic trajectories. This article presents a case study of inference mechanism on semantic trajectories. We propose a solution based on an ontological approach for modelling semantic trajectories integrating time information and rules. We give experiments and evaluations of the proposed approach on generated and real data.

1 Introduction

Over the last few years, there has been a huge collection of real-time data of mobile objects. These data are obtained by satellite based systems like GNSS¹, GPS² or ARGOS, phone location or radio-frequency identification. This opens new perspectives for several applications like road traffic supervision and animals tracking. Therefore, it becomes necessary to provide mechanisms for storage, modelling, efficient analysis and knowledge extraction from these data. The raw data captured,

Rouaa Wannous · Jamal Malki · Alain Bouju
University of La Rochelle, L3i Laboratory, EA 2118, F-17000 La Rochelle, France

Cécile Vincent
University of La Rochelle, LIENSs Laboratory, UMR 7266, F-17000 La Rochelle, France
e-mail: {rwanno01,jmalki,abouju,cvincent}@univ-lr.fr

¹ GNSS : Global Navigation Satellite System.

² GPS: Global Positioning System.

commonly called trajectories, traces moving objects from a departure point to a destination point as sequences of pairs (sample points captured, time of the capture). In [12], authors give a general definition of a trajectory: “A trajectory is the user defined record of the evolution of the position (perceived as a point) of an object that is moving in space during a given time interval in order to achieve a given goal”. Trajectories can be constrained to existing networks [10], or be unconstrained like in our study. Raw trajectories don’t contain contextual information about moving objects like goals of travelling nor activities accomplished [3]. To consider these semantic information, semantic trajectories are defined as a result of the annotation process of raw data with semantic annotations [12]. This annotation process can be done automatically or manually. Semantic trajectories can be seen as a high-level information layer on raw trajectories [14]. In [8], to model semantic trajectories, a domain ontology is constructed to represent domain concepts and rules. In the continuation of this paper [8], we discuss strategies for time integration with evaluation on synthetic and real data. We study seal trajectories and focus on semantic annotations for these activities such as foraging, travelling and resting. The inference mechanism on semantic trajectories which is connected to time knowledge has time and space storage complexity problem. This work addresses these two problems and gives some ideas for improving the complexity of the proposed approach.

This paper is organized as follows: section 2 presents the state of the art on semantic trajectories and some recent related work. Section 3 details our domain application and queries we aim to answer. Section 4 gives the two ontologies needed, seal trajectory and time ontologies. Section 5 presents our domain ontology rules and the temporal ontology rules. Section 6 defines the connection between trajectory and time ontologies. Section 8 discusses the evaluation of the proposed approach. Finally, section 9 concludes this paper and presents ideas for the future work.

2 Related Work

Data management techniques including modelling, indexing and querying large spatio-temporal data are actively investigated during the last decade [13]. Most of these techniques are only interested in raw trajectories. Projects like GeoP-KDD [4] and MODAP³ emphasized the need to address and to use semantic information about moving objects for efficient trajectories analyses. Recently, new projects are born like MOVE⁴ who aims to improve methods for knowledge extraction from massive amounts of moving objects data. As an example, in birds migration project [12], trajectories are analysed for better understanding birds behaviours. Scientists try to answer queries such as: where, why and how long birds stop on their travels, which activities they do during their stops, and which weather conditions the birds face during their flight. Considering these new requirements, new researches

³ MODAP: Mobility, Data Mining and Privacy.

⁴ Move: European Cooperation in Science and Technology - <http://move-cost.info/>

have emerged offering data models that can easily be expanded to take into account semantic data. In [12], the trajectory is seen as a user defined time-space function from a temporal interval to a space interval. To consider semantics of trajectories, a conceptual view is defined by three main concepts: stops, moves, and begin-end of a trajectory. Each part contains a set of semantics data. This model is implemented and evaluated on a relational database. Most domain and temporal operations are SQL based and use elementary data comparators. Based on this conceptual model of trajectories, several works have been proposed such as [3].

Using ontologies for semantic spatio-temporal data modelling is a new research field. In [9], authors work on a military application domain with complex queries that require sophisticated inferences methods. For this application, they present an upper-level ontology defining a general hierarchy of thematic and spatial entity classes and associating relationships to connect these entity classes. They intend for application-specific domain ontologies in the thematic dimension to be integrated into the upper-level ontology through subclassing of appropriate classes and relationships. Temporal information is integrated into the ontology by labelling relationship instances with their valid times. In this work, the temporal and spatial dimensions are included in the global ontology. Moreover, the ontology is formalised by the RDFS vocabulary and implemented on a relational database. Consequently, the inference mechanism is based on several domain specific table functions. The inference mechanism defined uses only the RDFS rules indexes. In [14], authors design a conceptual model of trajectories based on the approach introduced in [12]. This model represents trajectories from low-level real-life GNSS data to different semantically abstracted levels. Their application concerns daily trips of employees from home to work office and coming back. So, they start from basic abstractions (e.g. stops, moves) to enriched higher-level abstractions (e.g. office, shop). In [6], Malki et. al define an ontological approach modelling and reasoning on trajectories. This approach takes into account thematic, temporal and spatial rules. The ontologies constructed are formalised using both RDFS and OWL vocabulary. The inference mechanism is based on rules defined as entailments.

Finally, in [12], domain or time inference mechanism are not proposed neither spatio-temporal data mining are not investigated. However, in this paper, we focus on time knowledge integration and use inference mechanisms on semantic trajectories based on the approach introduced in [6]. Nevertheless, authors did not address the evaluation of the proposed approach. For all of that, this article gives experiments and evaluates the performance problems of time integration on generated and real data.

3 Domain Application

In this section, we present the seal trajectory modelling approach. We introduce the seal trajectory data model and its semantic associated layer.

3.1 Seal Trajectory Data Model

As in [6], this paper considers trajectories of seals. The data comes from the LIENSS⁵ (CNRS/University of La Rochelle) in collaboration with SMRU [11]. These laboratories work on marine mammals ecology. Trajectories of seals between their haulout sites along the coasts of the English Channel or in the Celtic and Irish seas are captured using GNSS systems provided by SMRU Instrumentation. We use trajectories data coming from GPS/GSM tags. The captured spatio-temporal data of seals trajectories can be classified into three main states: haulout, cruise and dive. The Fig. 1 shows the three states, the transitions and their guard conditions [8].

3.2 Semantic Seal Trajectory

We focus on studying seals' activities in order to identify their foraging areas. The main activities of seal, like foraging, resting and travelling, occur in the dive parts of the trajectory. So we aim at answering queries, such as:

1. foraging activities;
2. foraging activities during a given time interval;
3. foraging activities performed after travelling during a given time interval.

For all queries, we have to define a domain rule called “foraging” on the seal trajectory model. However, for the last two queries, time rules must be defined between trajectory's parts. For example, the query 3 needs the two domain rules “foraging, travelling” and the time rule “during” as illustrated by the Table 1

4 Modelling Approach

The need of a time model with temporal relationships appears in Table 1. As in [8], we consider separated and independent data models using an ontological approach.

4.1 Seal Trajectory Ontology

The seal trajectory ontology, owlSealTrajectory, is a result of a transformation model of the semantic seal trajectory. An extract of this ontology is in Fig. 2. This ontology defines the main following concepts:

- Seal: represents the animal equipped with a tag;
- Sequence: captures in the form of temporal intervals with a spatial part called GeoSequence and can be Haulout, Cruise or Dive. The other parts are metadata called Summary and CTD (Conductivity-Temperature-Depth);

⁵ <http://lienss.univ-larochelle.fr>

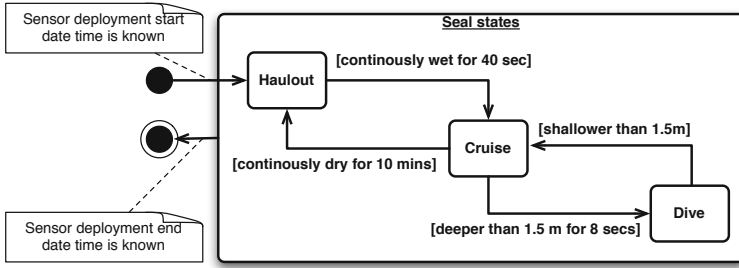


Fig. 1 The three states of seal trajectory

Table 1 Domain and time concepts and rules needed by the query 3

Concepts and rules		Description	
Concepts	Domain	Dive	specific part of the seal trajectory
	Time	Temporal-interval	the given temporal interval
Rules	Domain	Travelling	seal activity
		Foraging	seal activity
	Time	After	temporal relationship between the two activities
		During	temporal relationship between activity and time interval

- Trajectory: is the logical form to represent a set of sequences;
- Activity: is the seal activity for a sequence or for a trajectory.

Besides these concepts, owlSealTrajectory defines these relationships:

- seqHasActivity: is the object property between an activity and a sequence;
- TAD (Time Allocation at Depth): is the data property calculated to define the shape of a seal’s dive, as mentioned in [7].

4.2 Time Ontology

Table 1 clearly highlights the need for temporal concepts as well as temporal relationships between these concepts. In our approach, we chose owlTime ontology [5] developed by the World Wide Web Consortium (W3C) thanks to the definition of the temporal concepts and relationships as defined by Allen algebra [1]. An extract of the declarative part of this ontology [6] is given in Fig. 3.

⁶ <http://www.w3.org/2006/time>

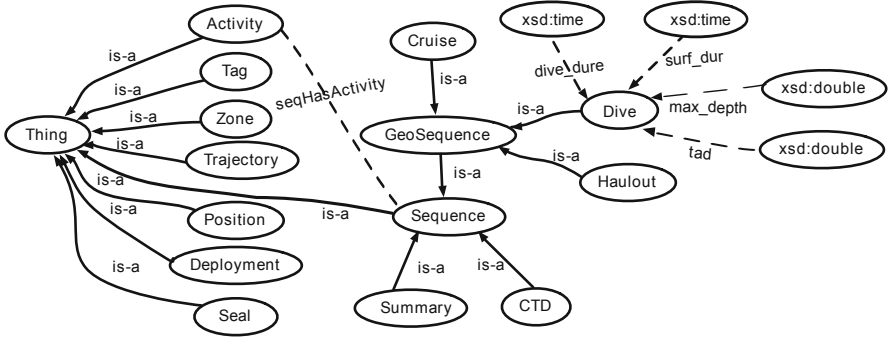


Fig. 2 An extract of the ontology owlSealTrajectory

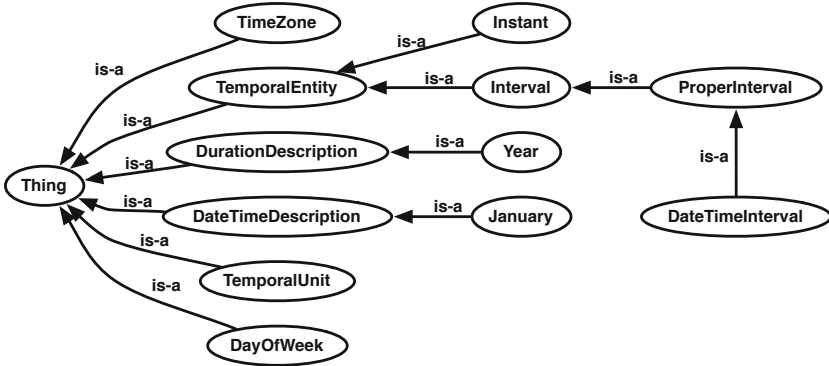


Fig. 3 A view of the owlTime ontology

5 Ontological Rules

5.1 Seal Trajectory Ontology Rules

We define four seals’ activities during dives: resting; travelling; foraging; travelling-foraging. In our approach, each seal activity is defined in the ontology and has both a declarative and an imperative corresponding parts. The Fig. 4 shows the declarative parts. The imperative parts are based on the decision Table 2 which determines the maximum dive depth, the dive’s shape (TAD) and the surface ratio dividing the surface duration by the dive duration. We implement the imperative parts using the Oracle database supporting semantic technologies. We create the rule base `sealActivities_rb` to hold the activities’ implementation as domain rules. The Code 1 gives the implementation of `foraging_rule` (line 3) in the rule base `sealActivities_rb`. In this code, the line 6 checks the maximum dive depth d to be more than 3 meters, the TAD t to be 0.9 and the surface duration s divided by the dive duration v , to be smaller than 0.5.

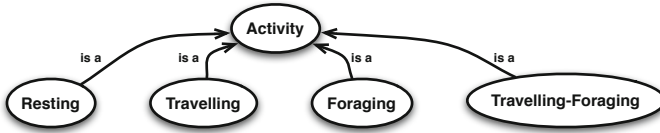


Fig. 4 Declarative part of seal activities

Table 2 Decision table associated to seal activities

Rules	Maximum dive depth	Dive shape or TAD	Surface ratio = surface duration/dive duration
Resting	< 10	all	> 0.5
Travelling	> 3	> 0 & < 0.7	< 0.5
Foraging	> 3	> 0.9 & < 1	< 0.5
Travelling_Foraging	> 3	> 0.7 & < 0.9	< 0.5

```

1 EXECUTE SEM_APIS.CREATE_RULEBASE('sealActivities_rb');
2 INSERT INTO mdsys.semr_sealActivities_rb
3 VALUES('foraging_rule',
4 '(?x rdf:type ost:Dive)
5 (?x ost:tad ?t) (?x ost:max_depth ?d) (?x ost:surf_dur ?s) (?x ost:dive_dur ?v)',
6 '(d > 3) and (t > 0.9) and (s/v < 0.5)',
7 '(?x ost:seqHasActivity ?activiteJ) (?activiteJ rdf:type ost:Foraging)',
8 SEM_ALIASES(SEM_ALIAS('ost', 'http://l3i.univ-larochelle.fr/Sido/
   owlSealTrajectory#')));
  
```

Code 1 The imperative part of the seal activity foraging

5.2 Time Ontology Rules

The owlTime ontology declares 13 relationships based on Allen algebra [1]. These are: intervalEquals, intervalBefore, intervalMeets, intervalOverlaps, intervalStarts, intervalDuring, intervalFinishes, intervalAfter, intervalMetBy, intervalOverlappedBy, intervalStartedBy, intervalContains, intervalFinishedBy. Allen temporal relationships are implemented inside the rule base owlTime_rb. For example, the Code 2 presents the implementation of the intervalAfter_rule.

```

1 EXECUTE SEM_APIS.CREATE_RULEBASE('owlTime_rb')
2 INSERT INTO mdsys.semr_owltime_rb
3 VALUES('intervalAfter_rule',
4 '(?x rdf:type owltime:ProperInterval) (?y rdf:type owltime:ProperInterval)
5 (?x owltime:hasEnd ?xEnd) (?xEnd :inXSDDateTime ?xEndDateTime) (?y owltime:
   hasBeginning ?yBegin) (?yBegin :inXSDDateTime ?yBeginDateTime)',
6 '(yBeginDateTime > xEndDateTime)',
7 '(?y owltime:intervalAfter ?x)',
8 SEM_ALIASES(SEM_ALIAS('owltime', 'http://www.w3.org/2006/time#')));
  
```

Code 2 The imperative part of Allen temporal relationship intervalAfter

6 Semantic Integration by Ontological Mapping

The need of a semantic integration clearly appears while considering separated and independent sources of information, like seal trajectory and time ontologies. This ontological mapping may lead to discover more semantic trajectory patterns. The property `rdfs:subClassOf` is not appropriate in separated ontologies. Even more the property `owl:sameAs` means that the two connected classes have the same intention meaning, however it does not go further for their properties. Consequently the property `owl:equivalentClass` is the most appropriate connection in our case. The mapping process is shown in Fig. 5 following these steps:

1. `owlSealTrajectory:Sequence` is the mapping concept by OWL construct `owl:equivalentClass` to `time:ProperInterval`;
2. `owlSealTrajectory:s_date` is the mapping object property by OWL construct `owl:equivalentProperty` to `owlTime:hasBeginning`;
3. `owlSealTrajectory:e_date` is the mapping object property by the OWL construct `owl:equivalentProperty` to `owlTime:hasEnd`.

In particular, the reasoner considers the owl property “`owl:equivalentClass`” which allows the inference of a “Sequence” instance as a “ProperInterval” instance. Therefore, the interval temporal rules are also valid for sequences of trajectories, which means valid for dives also.

7 Temporal Rule Extension

The inference mechanism is needed for queries on the semantic trajectory `owlSealTrajectory` mapped to the time ontology `owlTime`. Calculating the inference between all sequences of trajectories considering all time rules takes a huge amount of time and space storage capacity. To enhance the inference mechanism, we define a refinement called **temporal neighbour inference**: “*A temporal neighbour is when a sequence happened within a conceptual distance to another*”. The goal of this refinement, algorithm 11 is to consider the distance between two sequences in order to calculate the corresponding temporal relationships. The temporal rules must comply with this refinement. It is still difficult to determine the best candidate for the temporal neighbour distance, and even then, there is uncertainty on its usefulness.

8 Evaluation and Analysis

The experiments aim at checking the usefulness of the temporal neighbour refinement. We create a synthetic temporal interval data and examine the validity of our proposal. Then we evaluate it’s efficiency on real GPS-GSM seal

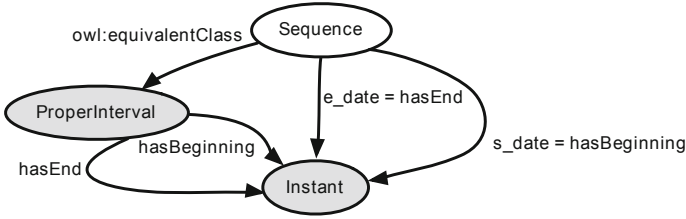


Fig. 5 Integration of owlSealTrajectory with owlTime ontology

```

input : Two sequences: a referent  $S_r$  and an argument  $S_a$ 
input : A neighbour of  $S_r$ 
output: Temporal rule between  $S_r$  and  $S_a$ 
if  $S_a \in$  to the neighbour of  $S_r$  then
  | calculate the temporal rule between  $S_r$  and  $S_a$ ;
  | go the next sequence  $S_a$ 
else
  | go the next sequence  $S_a$ 
end

```

Algorithm 1. Temporal neighbour inference algorithm

trajectory data. We use Oracle 11g Release 2. For the experiments, we consider the query 3 (§3.2). The Code 3 gives the SQL code of this query.

```

1 SELECT D1, D2
2 FROM TABLE ( SEM_MATCH(
3 '(?D1 rdf:type ost:Dive) (?D1 ost:sequenceHasActivity ?activiteD1) (?
   activiteD1 rdf:type ost:Travelling)
4 (?D2 rdf:type ost:Dive) (?D2 ost:sequenceHasActivity ?activiteD2) (?
   activiteD2 rdf:type ost:Foraging)
5 (?D1 ot:intervalBefore ?D2)
6 (?I rdf:type ot:ProperInterval) (?I ot:hasBeginning ?beginI) (?beginI ot:
   inXSDdateTime "2003-08-02T00:00:00"^^xsd:datetime) (?I ot:hasEnd ?endI
   ) (?endI ot:inXSDdateTime"2003-08-09T23:59:00"^^xsd:datetime)
7 (?D1 ot:intervalDuring ?I) (?D2 ot:intervalDuring ?I)',
8 SEM_Models('owlSealTrajectory','owlTime'),
9 SEM_Rulebases('OWLPRIME','sealActivities_rb','owlTime_rb'),
10 SEM_ALIASES(SEM_ALIAS('ost', 'http://13i.univ-larochelle.fr/Sido/
   owlSealTrajectory#'),
11 SEM_ALIAS('ot','http://www.w3.org/2006/time#')),
12 null));

```

Code 3 The SQL code of the query 3

In Fig. 6 the inference mechanism is done on semantic seal trajectory before and after the mapping. The experiment is done for different numbers of dives shown in the horizontal axis multiple by 100. In Fig. 6(a), the vectorial axis shows the time multiple by 10 000 needed for the inference mechanism. In Fig. 6(b), the vectorial axis shows the number of triples multiple by 100 000 related to the space storage. From analysing Fig. 6 the problem is obvious comparing the time and the space storage needed from the inference mechanism. In other words, after using temporal rules, calculating the inference becomes very expensive

in terms of time and the space storage. In our point of view, this problem is related to the temporal rules integration without any constraints. With biological feedback, we define the temporal neighbour distance for seal trajectories to five minutes (300 seconds). Then we modify the implementation of temporal rules considering the temporal neighbour refinement. For instance, the modification of the temporal `intervalAfter` rule with the temporal neighbour refinement, called `intervalAfterRefined` rule, is given by the Code 4.

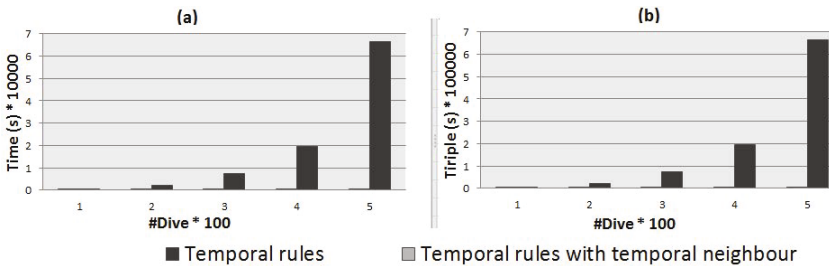


Fig. 6 Compare the time and space storage taken from the inference mechanism on the semantic seal trajectory integrated with/without the temporal rules

```

1 EXECUTE SEM_APIS.CREATE_RULEBASE('owlTime_rb')
2 INSERT INTO mdsys.semr_owltime_rb VALUES(
3 'intervalAfterRefined_rule',
4 '(?x rdf:type owltime:ProperInterval) (?x owltime:hasEnd ?xEnd) (?xEnd :
   inXSDDateTime ?xEndDateTime) (?y rdf:type owltime:ProperInterval) (?y
   owltime:hasBeginning ?yBegin) (?yBegin :inXSDDateTime ?yBeginDateTime)'
5 '(yBeginDateTime > xEndDateTime) and ((timeIntervalLengthInSeconds(
   dateTime2TimeStamp(xEndDateTime),dateTime2TimeStamp(yBeginDateTime))
   <300)',
6 '(?y owltime:intervalAfterTime ?x)',
7 SEM_ALIASES(SEM_ALIAS('owltime','http://www.w3.org/2006/time#'));

```

Code 4 Create the temporal `intervalAfterRefined_rule`

Then we validate the usefulness of the temporal neighbour on the synthetic data integrated first with the temporal rules and later with the extended temporal rules, as shown in Fig. 7. The vectorial axis, Fig. 7(a) shows the time and Fig. 7(b) shows the number of triples, where both time and space needed for the inference mechanism. Figure 7 shows the useful impact of applying the refined temporal rules in reducing the time and the space storage. Finally we apply the experiment on a real GPS/GSM data integrated first with the temporal rules and then with the refined temporal rules, as shown in Fig. 8. In Fig. 8(a), the vectorial axis shows the time (multiple by 10 000) needed for the inference mechanism for each number of dive. In Fig. 8(b), the vectorial axis shows the number of triples (multiple by 100 000) related to the space storage taken from the inference mechanism for each number of dive. Figure 8 shows the improvement made on calculating the inference after applying the temporal neighbour concept. In

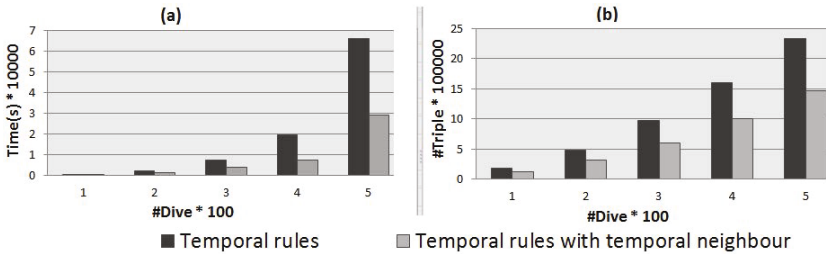


Fig. 7 The contribution of using temporal neighbour reduced the time and space storage taken form the inference mechanism on the synthetic data

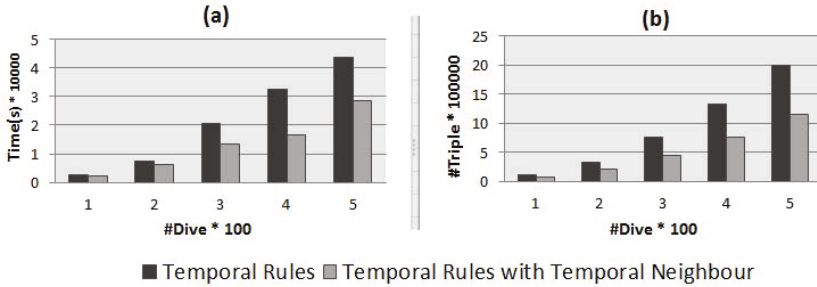


Fig. 8 The impact of using temporal neighbour refinement in the time and space storage taken by the inference mechanism on the GPS/GSM data

other words, the temporal neighbour is efficient in reducing the complexity of the inference mechanism on the GPS/GSM data.

9 Conclusion and Future Work

Trajectory data are usually available as sample points, and lake of semantic information, which is of fundamental importance for the efficient use of these data [2]. In this paper, we present a case study on the use of an ontological based approach for modeling semantic trajectories integrated with time rules. The main goal is to apply the inference mechanisms on semantic trajectories and to present a solution to reduce the complexity reasoning and querying on these semantic trajectories. We define a new condition on temporal rules which is *temporal neighbour refinement*. Then, we evaluate our approach on synthetic data as well as on real GPS/GSM data. The experiment’s results verify the positive impact of the temporal neighbour refinement on reducing the complexity of the inference mechanism. The influence of one condition positively appears nevertheless the reasoning complexity still exists. So applying more domain conditions on rules is therefore very important for reducing time and space storage inference complexity. As future work, we aim at applying more conditions on

temporal and spatial rules. We also intend to assess the impact of using incremental inference to improve the inference mechanism complexity and the contribution of temporal and spatial relationships composition.

References

1. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11), 832–843 (1983)
2. Alvares, L.O., Fernandes, J.A., Macedo, D., Bogorny, V., Moelans, B., Kuijpers, B., Vaisman, A.: A Model for Enriching Trajectories with Semantic Geographical Information. In: *ACM-GIS*. ACM Press (2007)
3. Baglioni, M., Macedo, J., Renso, C., Wachowicz, M.: An Ontology-Based Approach for the Semantic Modelling and Reasoning on Trajectories. In: Song, I.-Y., Piattini, M., Chen, Y.-P.P., Hartmann, S., Grandi, F., Trujillo, J., Opdahl, A.L., Ferri, F., Grifoni, P., Caschera, M.C., Rolland, C., Woo, C., Salinesi, C., Zimányi, E., Claramunt, C., Frascinar, F., Houben, G.-J., Thiran, P. (eds.) *ER Workshops 2008*. LNCS, vol. 5232, pp. 344–353. Springer, Heidelberg (2008)
4. GeoPKDD. Geographic Privacy-aware Knowledge Discovery and Delivery. Coordinator: KDDLAB, Knowledge Discovery and Delivery Laboratory, ISTI-CNR and University of Pisa, <http://www.geopkdd.eu>
5. Hobbs, J.R., Pan, F.: An Ontology of Time for the Semantic Web. *ACM Transactions on Asian Language Information Processing* 3, 66–85 (2004)
6. Malki, J., Bouju, A., Mefteh, W.: An Ontological Approach Modeling and Reasoning on Trajectories. Taking into account Thematic, Temporal and Spatial Rules. *Revue des Sciences et Technologies de l'information* 31(1), 71–96 (2012), doi:10.3166/tso.31.71-96
7. Jonsen, I.D., Myers, R.A., James, M.C.: Identifying Leatherback Turtle Foraging Behaviour from Satellite Telemetry using a switching State-space Model. *Marine Ecology Progress Series* 337, 255–264 (2007)
8. Malki, J., Mefteh, W., Bouju, A.: Une Approche Ontologique pour la Modélisation et le Raisonnement sur les Trajectoires. Prise en compte des règles métiers, spatiales et temporelles. In: *JFO 2009 3ème édition des Journées Francophones sur les Ontologies*, France, pp. 157–168 (December 2009)
9. Perry, M.: A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data. PhD thesis, Wright State University (June 10, 2008)
10. Sandu Popa, I., Zeitouni, K., Oria, V., Barth, D., Vial, S.: Indexing in-Network Trajectory Flows. *The VLDB Journal* 20(5), 643–669 (2011)
11. SMRU. Sea Mammal Research Unit. Collaborative ventures between marine biologists and systems engineers, University of St. Andrews, UK, <http://www.smru.st-and.ac.uk/>
12. Spaccapietra, S., Parent, C., Damiani, M., Demacedo, J., Porto, F., Vangenot, C.: A Conceptual View on Trajectories. *Data and Knowledge Engineering* 65(1), 126–146 (2008)
13. Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., Aberer, K.: SeMiTri: A Framework for Semantic Annotation of Heterogeneous Trajectories. In: *Proceedings of the 14th International Conference on Extending Database Technology, EDBT/ICDT 2011*, pp. 259–270. ACM, New York (2011)
14. Yan, Z., Parent, C., Spaccapietra, S., Chakraborty, D.: A Hybrid Model and Computing Platform for Spatio-semantic Trajectories. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) *ESWC 2010*. LNCS, vol. 6088, pp. 60–75. Springer, Heidelberg (2010)

WebOMSIE: An Ontology-Based Multi Source Web Information Extraction

Zineb Younsi, Mohamed Quafafou,
Redouane Ouzegane, and Abdelkamel Tari

Abstract. The Web contains a huge volume of information supplied by diverse sources such as e-commerce sites, electronic directories, search engines, etc. The difficulty of the task of automating information extraction from these sources lives on the fact that these last ones were conceived for a human access (manual navigation). This difficulty is increased as the number of sources in question increases. In this paper, we are interested in the problem of EI, from several sources. The first approach to resolve this problem consists in suggesting a new method of EI and applying it to the various sources. This approach is not very successful and it is difficult to implement, especially when the sources are very heterogeneous. Therefore, We propose a more effective alternative, allowing us to benefit from already existing methods and tools, by applying to every source, the tool which suits most. For that purpose, we exploit domain ontology to deduct the tool adequate to every source. In this paper, we present the WebOMSIE system, an ontology-based framework of multi source information extraction from the Web.

Keywords: Information extraction, *WETDL*, ontology, knowledge base, reasoner, descriptive logic.

1 Introduction

The aim of the information extraction (IE) is to automatically extract from a semi-structured or not structured document, information readable by a

Zineb Younsi · Redouane Ouzegane · Abdelkamel Tari

Bejaia University, Bejaia 06000 Algeria

e-mail: [zineb.younsi, redouane.ouzegane}@univ-bejaia.dz](mailto:{zineb.younsi, redouane.ouzegane}@univ-bejaia.dz)
abdelkamel.tari@univ-bejaia.dz

Mohamed Quafafou

Marseille University France, Saint Jérôme Marseille

e-mail: mohamed.quafafou@univmed.fr

machine. This activity involves documents containing free text written in a natural language, for which we use natural language processing (NLP) techniques. NLP leans on the lexical analysis of a not structured document and on the grammatical structure of the sentences. IE may also involve sources consisting of multimedia documents i.e containing graphic, audio and/or video components, for which we use the automatic techniques of annotation and of concept extraction to extract information. The problem of IE is different from that of the Web information extraction (WIE). The latter deals only with semi-structured documents, that are pervasive on the Web, such as tables or itemized and enumerated lists (e.g. Fig. 1). Throughout this paper and except in case of opposite precision IE means WIE.

The IE problem is increased when we have to extract information from several sources. This paper describes WebOMSIE, an alternative ontology-based information extraction from multiple web sources system. We can find a big amount of IE approaches in literature, but each one is different of the other one according of its specificities, like the page type of the web source, the features used by the tool, the extraction rule type, the learning Algorithm used, etc. Instead of developing a new approach of IE to extract relevant information from several sources, WebOMSIE exploits a knowledge contained in an ontology to deduct which approach we will apply to extract from each source. Before exploiting the ontology, a user of the system has to precise for each source its information extraction criteria (IEC) and *WETDL* network. An extraction criterion can concern both of the source like, for example, Page Type(structured, semi-structured or free text) of the input documents that each IE system targets, or the tool itself, like the extraction rule that is used in the extraction process or the learning algorithm, etc. According to a source IE criteria, WebOMSIE infer in the ontology to deduct the approach tool to



Fig. 1 A Semi-structured page containing data records (in rectangular box) to be extracted

apply for each source in the information extraction process. Hence, WebOMSIE encapsulates for each source both of parameters (URL, page number...), IEC, the tool inferred by the ontology and the *WETDL* network in an extraction project. The execution of this later, consist to perform in parallel each tool on the corresponding source by following the *WETDL* Network.

The next section summarizes related work in information extraction area. Section 3 summarizes WebOMSIE architecture, while Section 4 describes our methodology of formal description and construction of our ontology. Section 5 describes what an information extraction project, which represent a model of multi source information extraction. Finally, we present in section 6 an implementation of our system, section 7 concludes.

2 Related Work on Information Extraction from the Web

The goal of the Web information extraction (WIE) is to build generic programs, which exploit the regularities of presentations and shaping of web pages to extract from it relevant information and put them in a structured and exploitable format by a machine. These programs are called Wrappers. Software tools used for the construction of adapters, called Wrapper induction systems (WI), take as parameters the regularities of presentation of web pages in the form of rules of extraction and generate a wrapper for this type of page. An extraction rule indicates to the wrapper how to find the information targets in a web page. Once found, the information must be saved in a data structure. Since few years, several classifications of WIE methods were proposed in the literature [1] [2] [3]. Habegger [7] [8] in its thesis proposes a classification of the methods of WIE according to the type of the data taken at input. He distinguishes then the manual approaches, the inductive semi-automatic approaches requiring a set of labeled examples pages, the structural approaches requiring only the raw pages, the approaches with knowledge bases, and the requiring approaches, only a set of example instances of a given relation. In his side, Chang and Al [1] suggest classifying EI systems according to the implication of the users, in four classes of approaches: manual (TSIMMIS [15], Minerva [23], WebOQL [5], W4F [20] and XWrap [24]), supervised (like SRV, RAPIER [22], WHISK [21], WIEN [18][19], STALKER [16], SoftMealy [14], NoDoSE [17] and DEByE [3]), semi supervised (IEPAD, OLERA [13] and Thresher [9]) and not supervised (RoadRunner [25], EX-ALG, DeLa and DEPTA [12]). We based our work on this last classification, for the suite of this paper. This choice is justified by the study and the comparison done by the authors of a great number of WIE methods according to several criteria. All approaches described previously rely on the structure of presentation features of the data within a document to generate rules or patterns to perform extraction. However, extraction can be accomplished by

relying directly on the data. Given a specific domain application, ontology can be used to locate constants present in the page and to construct objects with them. The most representative tool of this approach named BYU is the one developed by the Birgham Young University Data Extraction Group [10].

3 Overview of WebOMSIE's Architecture

The choice of an IE approach to apply for a web source often causes a problem to the user who has to take into account several aspects relative to the nature of this source and the requirements of the approach tool itself. For that purpose, we propose WebOMSIE: a system of multi source web information extraction by exploiting knowledge contained in the ontology OIE of IE methods met in literature.

3.1 *WebOMSIE Components*

WebOMSIE contains four main components, described as follows:(1) A graphic user interface (GUI): it allows the system user to enter sources information's like (parameters and EI criteria).(2) A descriptive logic (DL) reasoner: it creates a knowledge base from the ontology OIE and so infer in this knowledge base. (3) The ontology OIE: it contains the tools of EI of Web, (See section4).(4) An extraction project execution engine: it allows from the information entered by the user through its GUI, to: - infer in its knowledge base to find the adequate tool for extraction of every source. - launch the EI on every source, by applying the tool of corresponding EI - Apply a mediator to integrate the data coming from all the sources.

3.2 *WebOMSIE Extraction Process*

The WebOMSIE system takes as input, as shown in Fig. 2, a set of n Web sources. The user introduces informations about these sources (e.g. URL, page number,) and the IE criteria corresponding. It works as follow:

Input web sources information, IEC and the global data schema.

From its GUI (graphic user interface) the user precise formations about sources which allows the system to download them and the IE criteria for each source. The user input also, the global data schema of future data issued from the multi source extraction.

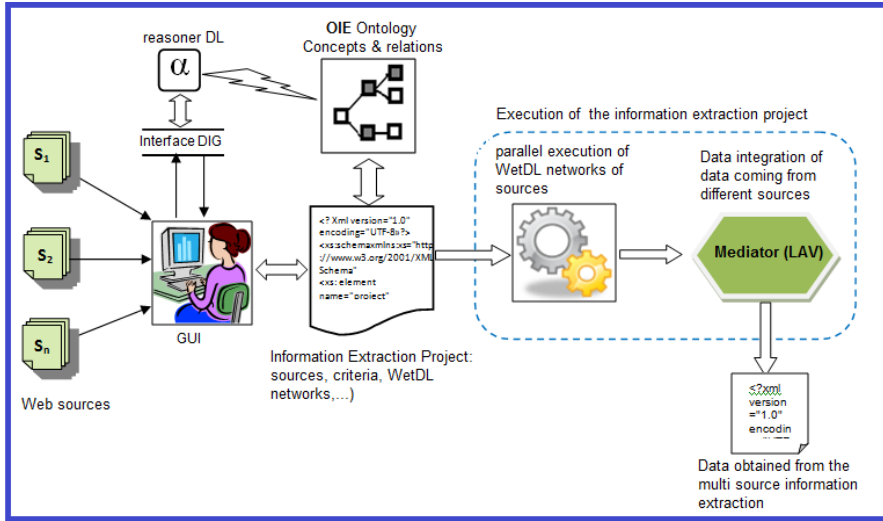


Fig. 2 The process of ontology-based multi source information extraction

Creation of an information extraction project. From the characteristics of sources inputted by the user (sources informations and IE criteria), the system creates a new IEP (see section 5).

Inference in the ontology OIE. From the information contained in *Criteria* element of each source of the IEP, the inference engine launches the reasoning in the ontology to deduce the tool adequate for the extraction from this source. So *Tool* element of every source are filled by the name of the tool to apply to it.

Launch of the multi source extraction. From the information contained in *Parameter* and *WETDL network*, the multi source extraction of the current IEP is launched. This is made by performing in parallel the execution of tool contained in the *Tool* element on every corresponding source. Finally, the *Data* element of every source is filled with the extracted data, in a structured format.

Integration of the data obtained from the sources. From the information contained in *Data* and *Local Schema* elements of each source, and the Global Schema contained in the current IEP, a process of integration by the approach *Local As View (LAV)* [4] is performed to reconcile the data coming from various sources. The *Data* element of the IEP is filled. The choice of the LAV approach for the data integration is justified by the ease to add or delete sources.

4 OIE Ontology Formal Description and Construction

In their works, Chang and Al [1] present a comparison of a great number of WIE systems according to three dimensions. The first one concerns the difficulty or the task domain that an IE task refers to, which can be used to answer the question: "why does not a system of IE manage to treat certain Web sites having a particular structure? ". The second dimension compares the underlying techniques used by the various systems of EI. The third dimension estimates the efforts supplied by the user during the process of learning, as well as the necessity of the EI system portability, through various domains. We exploited the comparison made by Chang and Al, and conceived and implemented a domain ontology that we call OIE which allows us to determine the WIE method that better suits to extract from the information from a given Web source.

4.1 Inference and Reasoning in OIE

The objective of the ontology OIE is to allow to determine from a set of criteria (seen in Chang comparison) which tool (or tools) it may verify them. More exactly, we associate to a given Web source a set of IE criteria and we want to know tools which can be applied to this Web source. For example, given a web source S, we want to know with which tools we can extract from source which has semi-structured page, and require programming by a user and which is manual tool? Then we specify three criteria: PageType , UserExpertise and hasClassif which have as value, respectively: "Semi-structured", " Programming" and "Manual".

Because OIE is made in descriptive logic (the logic SHOIN), we are forced to use a DL (Descriptive Logics) reasoned, which infers with the formalisms of the descriptive logic.

4.2 DIG Requests to OIE

To request the knowledge contained in OIE, we have used a reasoner DL (Description Logic) which support the DIG (Description Logics Implementation Group) interface [6]. Requests are expressed in DIG language called Asks. To explain the request send, we take the previous section example. This request can be expressed in Descriptive Logic as:

$\text{Tool} \cap \exists \text{hasClassif.} \text{"Manual"} \cap \exists \text{hasPageType.} \text{"semi-Structured"} \cap \exists \text{hasUserExpertise.} \text{"Programming"}$.

The DIG reasoner answers to this request by a set of names of tools which correspond to the specified criteria. This set is Minerva, TSIMMIS, WebOQL.

5 Information Extraction Project

Before presenting the IE project notion, we must present IEC , *WETDL* network and web source notions.

5.1 Information Extraction Criterion

An information extraction criterion (IEC) can be defined as a characteristic of a web source or an IE tool. For example (as presented in Sect. 4.2), *PageType* is a source criterion, *UserExpertise* and *hasClassif* are source criteria. Those criteria can have as value respectively: "Semi-structured", "Programming" and "Manual".

5.2 WETDL Network

To resolve the problem of web information extraction, it is not only enough to generate adapters for on-line sources. Besides this essential task, we have to specify also how to realize the other tasks as, the interrogation of the on-line sources (that is: the construction of the requests according to the language of every source)the download of pages results, the data mining from downloaded pages by applying an adapter, the selection of the format of extracted data (documents XML, relational tables, etc.), and possibly, in the case of IE from multiple sources, how to assure the aggregation and the integration of the data stemming from several sources in the same format. So, the stage of induction or generation of adapters for on-line sources constitutes only one under task of the applications of WIE. Habegger [8] proposes a solution bto this problem, which consists in decomposing the application or the IE task of elementary sub-tasks of IE, which are described by generic and customizable operators. The effective realization of the application consists in defining coordination between these operators. To describe and coordinate these operators, Habegger defines an XML language named *WETDL* (Web Extraction Task Description Language). So, this language leans on the notion of generic basic operators: query, fetch, parser, extract, transform, filter and external. Consequently, to describe a task of extraction of information in *WETDL* it is at first necessary to decompose it into sub- tasks then identify the corresponding operators in every sub-task. Once these operators were defined and parameterized for a task, it is necessary to coordinate them. Every operator receives data, applies them a treatment and sends a list of results which will become input data for one or several operations in the processing chain. This allows us to represent the IE execution plan by operators' network describing this task, where every operator will describe a sub-task of the extraction, this network is called "*WETDL network*".

5.3 Web Source

A web source (S_i) $i=1,n$ can be formalized by the sextuple:

$\langle (P_{ij}, VP_{ij}), (C_{ik}, VC_{ik}), (T_{ih}), (N_i), (LC_i), (D_i) \rangle$ $j=1, 1 ; k=1, m ; h=1, p$. Where,

- n : the number of the web sources to request;
- l : the number of parameters of the source S_i ;
- m : the number of criteria of the source S_i ;
- p : the number of tools that are adequate with source S_i .

Parameter element $(P_{ij}, VP_{ij})_{j=1,1}$. P_{ij} represents the j^{th} parameter of the source S_i . VP_{ij} represents the value of the j^{th} parameter of the source S_i . *Criteria* element $(C_{ik}, VC_{ik})_{k=1,m}$. C_{ik} is the k^{th} criterion of the source S_i . VC_{ik} is the value of the k^{th} criterion of the source S_i . *Tool* element $(T_{ih})_{h=1,p}$. T_{ih} is the h^{th} tool applicable to the source S_i . *WETDL network* element $(N_i)_{i=1,n}$ is the WETDL network of the source S_i . *Local schema* element $(L_{S_i})_{i=1,n}$. L_{S_i} is the local data schema of the source S_i . *Data* element $(D_i)_{i=1,n}$. T_{ih} corresponds to the data, expressed in a structured format, obtained by IE from the source S_i .

To formalize the problem of multi source web information extraction, we introduce the *information extraction project* (IEP). Formally, An IEP P can be defined a the quadruplet: $\langle S, O, GS, D \rangle$ Where,

S is a set of n Web sources; every source as defined in section 5.3. O is an ontology which associate for each source S_i ($i=1,n$) a list of tools $(T_{ih})_{h=1,p}$. In our work the ontology O correspond naturally to the OIE ontology, GS is the global schema of the data obtained after the multi source extraction and D is the data obtained from the multi source extraction, in a structured format.

6 Implementation

We have implemented the WebOMSIE system in the form of application Java. For this we have used the Java language which allows a big flexibility and portability for the application. We also implemented our ontology OIE under the editor Protege 2000. The inference on OIE is done with the reasoner Pellet [11].

7 Conclusion

In this paper, we have proposed an ontology based approach for a multi source Web information extraction problem. We have presented the ontology OIE as a solution to find adequate WIE tool to a given web source problem.

The presented approach brings a certain intelligence in the process of information extraction, but also more relevance with the use of the ontology to find the appropriate tool to extract relevant data from a given web source. We have developed an experimental prototype for the proposed approach. The next stage consists in estimating and evaluating this approach with regard to the traditional techniques of IE.

The architecture is a generic approach for interactive construction of IEP. According to the user (usually is not an expert in IE processing) requirements we have developed user-friendly interface that offers the possibility to label, add, remove, and parameterize the sources contained in the IEP. The designed IEP can then be stored and modified during batch processing.

The originality of our approach lies in the opportunity offered to the users to be able to build, in an interactive way, a multi source IE application. Furthermore, our approach is very interesting because of the generic version presented to the user at the creation of a new IEP where he can obtain an executable IEP by only parameter the application. Furthermore, the use of the language *WETDL* facilitates the writing of the sub-tasks of an IE task, which are implemented under the *WETDL* operators. Finally, the proposed approach is toward data integration by using LAV approach. This make easier the operation of removing or adding sources by expressing the local schema according to the global schema.

References

1. Chang, C.H., Kayed, M., Moheb, R.G., Shaalan, K.: A Survey of Web Information Extraction Systems. *IEEE Transactions on Knowledge and Data Engineering* 18 (2006)
2. Laender, A.-H.-F., RibeiroNeto, B.-A., da Silva, A.-S., Teixeira, J.-S.: A Brief Survey of Web Data Extraction Tools. *SIGMOD Record* (2002)
3. Laender, A.-H.-F., RibeiroNeto, B.-A., da Silva, A.-S.: DEByE - Data Extraction by example. *Data and Knowledge Engineering* (2001)
4. Lenzerini, M.: Data Integration: A Theoretical Perspective. In: *Symposium on Principles of Database Systems* (2002)
5. Arocena, G.O., Mendelzon, A.O.: WebOQL: Restructuring Documents, Databases, and Webs. In: *Proc. 14th IEEE Int'l Conf. Data Eng.*, pp. 24-33 (1998)
6. Bechofer, S.: The DIG Description Logic Interface: DIG/1.1. University of Manchester (2007)
7. Habegger, B.: Multi-Pattern Wrappers for Relation Extraction from the Web. In: *Proceedings of the European Conference on Artificial Intelligence* (2002)
8. Habegger, B.: Extraction d'informations partir du Web. Phd thesis Nantes University (2004)
9. Hogue, A., Karger, D.: Thresher: Automating the Unwrapping of Semantic Content from the World Wide. In: *Proc. 14th Int'l Conf. World Wide Web*, pp. 86-95 (2005)

10. Embley, D.-W., Campbell, D.-M., Jiang, Y.-S., Liddle, S.-W., Lonsdale, D.-W., Ng, Y.-K., Smith, R.-D.: Conceptual-Model-Based Data Extraction from Multiple-Record Web pages. *Data and Knowledge Engineering* 31, 227–251 (1999)
11. Bijan Parsia, B., Evren, S.: Pellet: An owl dl reasoned. In: *Proceedings of the International Workshop on Description Logics* (2004)
12. Wang, J., Lochovsky, F.H.: Data Extraction and Label Assignment for Web Databases. In: *Proc. 12th Int'l Conf. World Wide Web (WWW)*, pp. 187–196 (2003)
13. Chang, C.-H., Lui, S.-C.: IEPAD: Information Extraction based on Pattern Discovery. In: *Proceedings of the ACM WWW 10 Conference* (2001)
14. Hsu, C.-N., Dung, M.-T.: Generating finite state transducers for semi-structured data extraction from the web. *Information Systems* 23, 521–538 (1998)
15. Hammer, J., McHugh, J., Garcia-Molina, H.: Semistructured Data: The TSIM-MIS Experience. In: *Proc. First East-European Symp. Advances in Databases and Information Systems* (1997)
16. Muslea, I., Minton, S., Knoblock, C.: Hierarchical Wrapper Induction for Semistructured Information Sources. *Autonomous Agents and Multi-Agent* 1 (2001)
17. Aderlberg, B.: NoDoSE: A Tool for Semi-Automatically Extracting Structured and Semi-Structured Data from Text Document. *SIGMOD Record* 27, 283–294 (1998)
18. Kushmerick, N.: Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence* 118, 15–68 (2000)
19. Kushmerick, N.: Finite-state approaches to web Information Extraction. In: *3rd Summer Convention on Information Extraction* (2002)
20. Sahuguet, A., Azavant, F.: Building intelligent web applications using lightweight wrappers. *Data and Knowledge Engineering* 36 (2001)
21. Soderlan, S.: Learning Information Extraction Rules for semi-Structured and Free Text. *Machine Learning* 34, 233–272 (1999)
22. Freitag, D.: Machine Learning for information Extraction in informal domains. *Machine Learning* 39, 169–202 (2000)
23. Crescenzi, V., Mecca, G.: Grammers Have Exceptions. *Information Systems* 23, 539–565 (1998)
24. Liu, L., Pu, C., Han, W.: XWRAP: An XML-enable Wrapper Construction System for web information Sources. In: *Proceedings of the 16th IEEE International Conference on Data Engineering*, pp. 611–621 (2000)
25. Crescenzi, V., Mecca, G., Merialdo, P.: RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In: *Proceedings of the 26th International Conference on Very Large Database Systems*, pp. 109–118 (2001)

Part IV
Modeling Multi-commodity Trade: Data
Models and Processing

Bidding Languages for Continuous Auctions

Mariusz Kaleta

Abstract. Bidding languages are well-defined for combinatorial auctions. However, the auctions of divisible goods are quite common in practice. In contrast to combinatorial auctions, in a case of divisible commodities, the feasible volumes of offers are continuous. Thus, we call such auctions as continuous auctions. In the paper we introduce three families of bidding languages for continuous auctions. They are based on the concepts derived from the combinatorial auctions. We generalize the language families based on goods, bids, and some mixture of both of them, to the continuous case. We also present preliminary analysis of their properties. Simple examples, reflecting the complementarity and substitutability, are provided with the exemplary representations in different languages.

1 Introduction

Market mechanisms are entering into new areas of life. From the beginning of this century one can observe rapidly increasing number of organized markets, mainly auctions, both on the retail and wholesale markets. Electronic auction mechanisms are visible in the Internet at specialized web pages, including these most recognizable, like eBay. But they are also being introduced into other web services, e.g. social networks, where availability of additional information enables the new functionalities. Trends to real-time operating are one of the main drivers for electronic trade on the wholesale markets. The more and more competitive conditions have led to the several management concepts like real-time enterprises or dynamic supply chain. Besides more competitive conditions, the nature of traded commodities forces the need of trading near the real time, that is, electronic markets are entering into markets of real-time commodities, e.g. telecommunication bandwidth, electrical energy, and so on. As a result of these trends, the new requirements and needs

Mariusz Kaleta
Warsaw University of Technology, Pl. Politechniki 1, 00-661 Warszawa
e-mail: mkaleta@ia.pw.edu.pl

for new functionalities are clear. The question how to built, integrate and manage the information systems for running the electronic auctions is addressed in a stream of researches [1, 4, 7]. Current research is focus on designing methodologies for auction systems. The reusability, extensibility, simplicity of deployment are the key aspects of the methodologies. Many works propose multi-agents approaches, e.g. [7], which make incentives for strong automation the trade. For example, in [1] a BPEL based approach is proposed to achieve flexibility in auction protocols.

One of the important problems in the context of electronic auction system designs is a bidding language [6]. A bidding language is a tool to represent valuations of a given agent to the market. From the point of view of particular agent, the overall evaluation of the market system strongly depend on the functionality that this tool gives to him. There are three criteria for evaluation of the bidding languages. The first is the expressive power, that is, what kind of valuations an agent can express using the particular bidding language. The second criterion is its succinctness – how verbose is a given language and as a result, how much memory it requires. And the last one is the simplicity and logical foundations of a given language from the agent point of view. A utility function of a given agent often has some logical structure. The question is, whether given language enables to exploit this structure in the bids. If so, the bids should be more convenient for the agents. Moreover, it usually lead to more succinct language. The succinctness is important for communication protocols and data management. Expressive, but simple languages may involve exponential growth of data describing a bid when a number of commodities increases. There is a wide stream of researches devoted to bidding languages for combinatorial auctions [2, 5, 6]. We discuss the achievements in this field further in the paper. However, relatively little attention in the literature has been paid to auctions of divisible goods, which are quite common in practice. We call such auctions as continuous auctions since the feasible volumes of bids are continuous in contrast to combinatorial auctions.

In the paper we focus on the bidding languages for the continuous auctions. We introduce the classes of bidding languages for the continuous auctions. In section 2 we introduce basic notions related to combinatorial and continuous auctions. Then, we describe bidding languages for combinatorial auctions in next section. Bidding languages for continuous auctions are introduced in section 4. After introducing the new classes of bidding languages we discuss their basic properties in section 5. We close the paper with a summary and indicating the directions of further research.

2 Auctions

On the ground of mechanism theory, an auction can be perceived as a mechanism. In a mechanism there is a set of agents who participate in certain game defined by the mechanism rules. Each agent does not reveal its private preferences, but instead

he sends the bids [4] to the mechanism. A bid encodes a valuation v that given agent has. Under certain conditions, the mechanism is triggered to compute a temporary market equilibrium and find the winners. After that the results are sent back from the mechanism to the agents.

Computation of temporary market equilibrium requires determining the volumes of winning bids and payments. Usually, it is done in two stages. First, the volumes of winning bids, and next, the market prices and related cash flows, are calculated. The problem of finding the winning bids is called the Winner Determination Problem (WDP).

In combinatorial auctions the agents may submit the bids on combinations of commodities. Assume, that $\mathcal{C} = \{1, 2, \dots, C\}$ is a set of commodities being traded. We also assume the following form of the valuation functions in combinatorial auctions: $v : 2^{\mathcal{C}} \rightarrow \mathbb{R}$. Later in this article we assume that v is normalized, which means that $v(\{\}) = 0$, and v is monotonic. Monotonicity means that if $X \subseteq Y$ then $v(X) \leq v(Y)$. We also assume the free disposal property which means that there is no cost of over allocation. The Winner Determination Problem is defined as follows.

Definition 1. (Winner Determination Problem, WDP) The seller has a set of commodities, $\mathcal{C} = \{1, 2, \dots, C\}$, to sell. The buyers submit set of offers (bids) $m \in \mathcal{B} = \{1, 2, \dots, B\}$. An offer m encodes valuation v_m . An allocation of commodities is denoted by $X_m(S) \in \{0, 1\}$, where $X_m(S)$ is equal to one if bundle $S \subseteq \mathcal{C}$ is allocated to the bid m . The Winner Determination Problem (WDP) is to find an allocation of commodities to buying offers which is revenue-maximizing under the constraints that no commodity is allocated more than once.

In the above, classical formulation of the WDP, it is assumed that each commodity can be allocated to at most one buying offer [5]. But in continuous case of WDP, each bid can be accepted partially, and commodities are perfectly divisible. Continuous WDPs are being solved in continuous auctions.

In the definition [1] there is nothing how the valuation is encoded in a bid. Offer encoding is the task of a bidding language. Without specifying a bidding language the description of market mechanism is not full and the market cannot be run.

3 Bidding Languages for Combinatorial Auctions

Three families of bidding languages for combinatorial auctions are under considerations in the literature [2]. The first family, denoted by \mathcal{L}_G , assumes that price and logical formula of commodities are provided in a bid. For a given allocation the formula can be evaluated to true or false. If it is evaluated to true, then the bid is accepted and paid at least the price given in the bid. Logical expression can be used instead of enumeration of all desired combinations of goods. If desired valuation for each combination is the same, then the expression can be used to build

¹ Although there are some subtle semantic differences in the following notions: *commodities* and *goods*, *bids* and *offers*, we treat them as synonymous in the paper.

one bid. Thus, \mathcal{L}_G allows to express agent preferences in natural way. In a case of perfect substitutability it exploits the logical structure of the preferences and leads to quite concisely bid. Exemplary language from the family \mathcal{L}_G is proposed in [3]. The authors have introduced the \mathcal{L}_G^{pos} language and its variants. \mathcal{L}_G^{pos} assumes that no negation can be used in the formulas. A disadvantage of \mathcal{L}_G is revealed when some combinations of goods have different valuations. In this case individual bids must be prepared.

In the second family of bidding languages, \mathcal{L}_B , the logical formulas of bids are provided. There are atomic bids, which are defined for a bundle of commodities and have associated prices. Any language from family \mathcal{L}_G combines the atomic bids in a logical formula. But in the contrast to \mathcal{L}_G , not whole formula is evaluated to true or false, but just individual atomic bids are checked to be satisfied or not. The price to be paid is a result of some function of atomic bid prices with respect to the logical formula of the bids.

An atomic bid is a tuple (\mathcal{G}, p) , $\mathcal{G} \subseteq \mathcal{C}$, where \mathcal{G} is a bundle of goods, and $p \in \mathbb{R}^+$ is a bid price. The atomic bid represents the following valuation:

$$v(X) = \begin{cases} p & \text{if } X \subseteq \mathcal{G} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Further, we will identified a bid with related valuation.

In \mathcal{L}_G^{or} language the logical operator OR is used to bind atomic bids. If two bids with valuations v_1 and v_2 are combined with OR operator, then the resulting valuation is as follows [6]:

$$(v_1 \text{ OR } v_2)(X) = \max_{X_1, X_2 \subseteq X, X_1 \cap X_2 = \emptyset} (v_1(X_1) + v_2(X_2)) \quad (2)$$

\mathcal{L}_G^{or} is not fully expressive, since it cannot represent any valuation with substitutability.

Another language from family \mathcal{L}_B , the XOR language \mathcal{L}_G^{xor} , is fully expressive. XOR combination of two valuations v_1 and v_2 defines the following valuation [6]:

$$(v_1 \text{ XOR } v_2)(X) = \max\{v_1(X), v_2(X)\} \quad (3)$$

Although XOR-bids can represent any valuations, it may need exponential number of atomic bids, while in the same case the OR language requires much less number of atomic bids.

Next three languages in \mathcal{L}_B family arise from the attempt to combine advantages of the two previously mentioned languages. In OR-of-XOR language a bid comprises OR combinations of XOR combinations of atomic bids. On the contrary in XOR-of-OR language there are XOR combinations of OR combinations of atomic bids. OR/XOR language is the most general since it allows for any combination of ORs and XORs. Each of these languages is fully expressive, but no one dominates in the meaning of their conciseness.

Nisan has also proposed a variant of OR language with phantom items – OR* language. An agent is allowed to include in his bids the phantom items which enable

to simulate XOR language. OR^* is fully expressive and it has also good properties in terms of conciseness, however it may require quadratic number of phantom items [6].

Boutilier and Hoos have proposed another family of bidding languages denoted by \mathcal{L}_{GB} . It allows for logical combination of both goods and bids and thus it inherits the advantages of both \mathcal{L}_G and \mathcal{L}_B families [2].

4 Bidding Languages for Continuous Auctions

Bidding languages for combinatorial auctions are well established in the literature. Thus, it is natural to derive the continuous languages from combinatorial ones. We will formulate languages for continuous auctions in relation to the languages defined in the previous section.

A valuation function for a continuous auction is a function $v : \mathbb{R}^C \rightarrow \mathbb{R}$. It is defined over the allocations space, where an allocation is a vector of commodities levels, $X = (X_c) \in \mathbb{R}^C, c \in \mathcal{C}, X_c \in \mathbb{R}$.

We assume, that each agent has its own utility function. Since the agents play a game and can act strategically, an agent uses a given language to show his valuation to the market. In general it may be different from his utility function. In combinatorial auction a bidding language can be used to present ones valuation in an approximately or accurately way. In the continuous case the accurate representation of the valuations would make the Winner Determination Problem too complex for efficient computation. In the rest of the paper we assume that the utility function, and so the valuations, are the lipschitz functions. We will focus only on bidding languages superfamily under the assumption that the valuations can be approximated by piecewise linear functions. As in combinatorial case in which one may enumerate all combinations and thus may present his utility accurately, also in continuous case an agent may achieve required error of approximation with sufficiently large number of pieces. If the number of pieces goes to infinity, then the approximation error converges to zero.

Definition 2. A bidding language is fully expressive in limit (fully expressive in short) if it can express any utility function using infinite (or less) number of linear pieces.

Computational complexity of a given language is a complexity related to determining the valuation of a bid in the language for a given allocation. The complexity of bidding language is important, because to make a decision about the allocation in the Winner Determination Problem, the value of the bids must be computed.

Definition 3. (taken from [6]) A bidding language is polynomially interpretable if there exists a polynomial time algorithm that for any bid in the language and given allocation X computes the value $v(X)$.

With a polynomially interpretable bidding language there is a hope to achieve efficient algorithm for the WDP. However, in the field of combinatorial auctions the

polynomially interpretable languages are not expressive enough and instead that, it is desired that having a proof (argument of the valuation function) it is possible to verify its optimality in polynomial time.

4.1 Family of Goods-Based Languages

Analogously to the family \mathcal{L}_G we introduce \mathcal{L}_{CG} a family of languages for continuous auctions. Instead of logical formulas on commodities, the domain $\mathcal{D} \subseteq \mathbb{R}^C$ of feasible allocations is provided in a bid in any language that belongs to the family \mathcal{L}_{CG} .

Definition 4. An offer in family \mathcal{L}_{CG} is a tuple (f, p, \mathcal{D}) , where

- $f : \mathbb{R}^C \rightarrow \mathbb{R}$ is a function to compute the normalized, unit volume,
- p is a price for a unit of normalized volume defined by function f ,
- $\mathcal{D} \subseteq \mathbb{R}^C$ is domain of feasible commodity allocation to the offer.

Then, the valuation is defined as follows:

$$v(X = (X_1, \dots, X_C)) = p\bar{x} \quad (4)$$

$$\bar{x} = \begin{cases} \max_{x \in \mathcal{D}, 0 \leq x_c \leq X_c} f(x) & \text{if there exists such } x \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Various languages in the family \mathcal{L}_{CG} differ in the way of definition of \mathcal{D} in a bid. If \mathcal{D} is a convex set then it can be defined by a simplex. Let us introduce a language $\mathcal{L}_{CG}^{simplex}$ from the family \mathcal{L}_{CG} with restriction that \mathcal{D} is defined by a simplex and $f(x)$ is linear. $\mathcal{L}_{CG}^{simplex}$ is polynomially interpretable since the valuation can be formulated as a linear programme.

Let us consider the following example: an agent would like to receive either c_1 or c_2 with total maximal volume equal to 4. For each unit of c_1 he is willing to pay 10, and for each unit of c_2 he is willing to pay 11. Notice, that this case cannot be represented in language \mathcal{L}_{CG} . But it can be represented in extended one, $\mathcal{L}_{CG}^{simplex*}$. Let $\mathcal{L}_{CG}^{simplex*}$ be the language $\mathcal{L}_{CG}^{simplex}$ with additional dummy commodities. These commodities are binary which means that each commodity can be accepted fully or not at all. Then, they can be used to model disjunctions like in the language \mathcal{L}_G^{OR*} [6]. In the above case, an agent must introduce dummy commodity c_3 . Then the offers may look like these: $\langle (c_1, 10, \{x : x_{c_3} \geq x_{c_1}/M\}) \rangle$, $\langle (c_2, 11, \{x : x_{c_3} \geq x_{c_2}/M\}) \rangle$, where M is huge enough number.

4.2 Family of Bids-Based Languages

Now, we will introduce the family \mathcal{L}_{CB} of languages which, similarly to \mathcal{L}_B , is based on function of atomic bids. There are two types of atomic bids: simple and bundle offers.

Definition 5. (Simple offer in continuous auction) Simple offer for a commodity $c \in \mathcal{C}$ is a pair (p, \mathcal{D}_c) , where p is an offer price, $\mathcal{D}_c \subseteq \mathbb{R}$ is feasible domain, that is, if the offer is winning, then the allocation X_c to this offer must satisfy the condition $X_c \in \mathcal{D}_c$.

The valuation of simple offer is defined as follows:

$$v(X = (X_1, \dots, X_C)) = \begin{cases} pX_c & \text{if } X_c \in \mathcal{D}_c \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Definition 6. (Bundle offer in continuous auction) Bundle offer is a tuple (α, p, \mathcal{D}) , where

- $\alpha = (\alpha_1, \dots, \alpha_C)$ is a vector of commodity shares, $\alpha_c \in \mathbb{R}$ is the share of commodity c in the bundle,
- p is an offer price of whole bundle,
- $\mathcal{D} \subseteq \mathbb{R}^C$ is feasible domain of commodities allocated to this offer.

The valuation of the bundle offer is defined as follows:

$$v(X = (X_1, \dots, X_C)) = p\bar{x} \quad (7)$$

where \bar{x} is an accepted volume of the bundle α :

$$\bar{x} = \begin{cases} \arg \max_{x \in \mathcal{D}, 0 \leq x_c \leq X_c} \min_c \left\{ \frac{x_c}{\alpha_c} \right\} & \text{if there exists such } x \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Simple offer is a special case of bundle offer, but sometimes it is convenient to refer to simple offers, especially if they are the only type of atomic bids that is allowed on a given auction. Notice, that in contrast to the combinatorial auctions, where limiting the atomic bids would make no sense, it could be quite natural in simple version of continuous auction.

In \mathcal{L}_{CB}^{or} language several atomic bids can be combined with the operator OR. Combining two bids, b_1 and b_2 , defines the following valuation:

$$(v_1 \text{ OR } v_2)(X) = \max_{(x)_1, (x)_2} (v_1((x)_1) + v_2((x)_2)) \quad (9)$$

subject to constraints

$$(x)_1 + (x)_2 \leq X, (x_i)_1, (x_i)_2 \geq 0 \quad (10)$$

where $(x)_1, (x)_2 \in \mathbb{R}^C$ and $(x_i)_n$ is i -th element of vector $(x)_n$.

In \mathcal{L}_{CB}^{xor} language two (or more) atomic bids can be combined with the operator XOR which defines the following valuation:

$$(v_1 \text{ XOR } v_2)(X) = \max_{(x)_1, (x)_2} \{v_1((x)_1), v_2((x)_2)\} \quad (11)$$

subject to constraints

$$(x)_1 + (x)_2 \leq X, (x_i)_1, (x_i)_2 \geq 0 \quad (12)$$

Analogously to the other languages defined in section 4.1, the languages $\mathcal{L}_{CB}^{or-of-xor}$, $\mathcal{L}_{CB}^{xor-of-or}$ and $\mathcal{L}_{CB}^{or/xor}$ can be also defined. Introduction an equivalent to OR* language requires that the dummy commodities are binary – they can be accepted with volume 1 or not accepted and this integrality must be taken into account in the equation (10).

4.3 Family of Goods- and Bids-Based Languages

The last family of languages is based on the concept of Boutilier and Hoos which is a kind of mixture of the previously defined families [2]. The language \mathcal{L}_{CGB} is defined as follows:

- bundle offer is in \mathcal{L}_{CGB} ,
- if $b_1, b_2 \in \mathcal{L}_{CGB}$ then $(b_1 \wedge b_2, p), (b_1 \vee b_2, p), (b_1 \oplus b_2, p)$ are all in \mathcal{L}_{CGB} .

Let $\Phi(b)$ be the formula associated with a bid b , one of the following formulas: $b, (b_1 \wedge b_2), (b_1 \vee b_2), (b_1 \oplus b_2)$. The function $\sigma(\Phi(b), X)$ gives the volume allocated to the bid b with formula Φ , when the allocation computed by WDP is X .

- If $\Phi(b)$ is a bundle offer, then $\sigma(\Phi(b), X) = \bar{x}$, \bar{x} is defined as in (5);
- If $\Phi(b) = b_1 \vee b_2$ or $\Phi(b) = b_1 \oplus b_2$, then $\sigma(\Phi(b), X) = \max(\sigma(\Phi(b_1), X), \sigma(\Phi(b_2), X))$;
- If $\Phi(b) = b_1 \wedge b_2$, then $\sigma(\Phi(b), X) = \min(\sigma(\Phi(b_1), X), \sigma(\Phi(b_2), X))$;

The valuation is defined as follows:

- If the bid is a bundle offer, then the valuation is equal to the one of bundle offer;
- If $\Phi(b) = b_1 \vee b_2$ then the valuation is a sum of valuations for $\Phi(b_1)$ and $\Phi(b_2)$ and $p\sigma(\Phi(b_1) \vee \Phi(b_2), X)$;
- If $\Phi(b) = b_1 \wedge b_2$ then the valuation is a sum of valuations for $\Phi(b_1)$ and $\Phi(b_2)$ and $p\sigma(\Phi(b_1) \wedge \Phi(b_2), X)$;
- If $\Phi(b) = b_1 \oplus b_2$ then the valuation is a sum of maximum of valuations for $\Phi(b_1)$ and $\Phi(b_2)$ and $p\sigma(\Phi(b_1) \vee \Phi(b_2), X)$;

Notice that in contrast to \mathcal{L}_{CB} there is no constraints like (10) or (12). So, if the allocation is satisfying many formulas, then ever formula is taken in the valuation. More justifications for the definition of language can be derived from its combinatorial version presented in [2].

5 Languages Properties

Languages $\mathcal{L}_{CB}^{simplex}$ and $\mathcal{L}_{CB}^{simplex*}$ are fully expressive since the utility can be represented by an infinite number of convex sets. $\mathcal{L}_{CB}^{simplex}$ is polynomially interpretable but it may require more bids than the $\mathcal{L}_{CB}^{simplex*}$.

From the family \mathcal{L}_{CB} only \mathcal{L}_{CB}^{or} is not fully expressive. Let us consider the following example. An agent can pay 2 for each unit of c_1 , 4 for each unit of c_2 , but if he gets both commodities he is willing to pay only 5. If the agent submits a bid $\langle (c_1, 2) OR (c_2, 4) OR (c_1 c_2, 5) \rangle$ in \mathcal{L}_{CB}^{or} and receives $X_{c_1} = X_{c_2} = 1$ then the value will be 6 instead of 5, which is wrong. This case is directly covered by \mathcal{L}_{CB}^{xor} language with the following offer: $\langle (c_1, 2) XOR (c_2, 4) XOR (c_1 c_2, 5) \rangle$. In fact, since \mathcal{L}_{CB}^{xor} enables to model disjunctive bids then it can be used to approximate any utility function with any accuracy. Thus \mathcal{L}_{CB}^{xor} , $\mathcal{L}_{CB}^{or-of-xor}$, $\mathcal{L}_{CB}^{xor-of-or}$, $\mathcal{L}_{CB}^{or/xor}$ are fully expressive.

$\mathcal{L}_{GB}^{simplex}$ and \mathcal{L}_{CB} naturally cover a case of additive valuations of goods. Let us redefine complementarity and substitutability in a case of divisible goods. We say that two goods are complementary if the valuation of both of them is greater than sum of valuations of individual goods. Two goods are substitution if the valuation of both of them is lower than sum of valuations of individual goods. Let us consider two examples related to the notions of complementary and substitutes.

Example 1. Suppose that an agent needs two complementary goods c_1 and c_2 with a joint value 10. The valuations for individual commodities are 1 and 2 for commodity c_1 and c_2 respectively. The maximal requested volume is 4. The required valuation is $10 * \min\{x_1, x_2\} + x_1 + 2x_2$, assuming that $x_1, x_2 \leq 4$.

In $\mathcal{L}_{CG}^{simplex}$ the above valuation can be represented by the following bids:

$$(f(x_1, x_2) = x_1, 10, \mathcal{D} = \{(x_1, x_2) : x_1 = x_2, 0 \leq x_1, x_2 \leq 4\}) \quad (13)$$

$$(f(x_1, x_2) = x_1, 1, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4\}) \quad (14)$$

$$(f(x_1, x_2) = x_2, 2, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_2 \leq 4\}) \quad (15)$$

In \mathcal{L}_{CB} language it can be defined as a set of atomic bids: $(13, \mathcal{D} = \{(x_1, x_2) : x_1 = x_2, 0 \leq x_1, x_2 \leq 4\})$, $(1, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4\})$, $(2, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4\})$.

In \mathcal{L}_{CGB} it can be represented in intuitive way, reflecting the structure of the utility function: $\langle ((1, 0), 1, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4\}) \wedge ((0, 1), 2, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_2 \leq 4\}) \rangle, 10$.

Example 2. Suppose that an agent needs two substitutable goods c_1 and c_2 . Each of them provides the basic valuation 10, but also each of them provides some additional bonus: 1, 2 in case of commodity c_1 and c_2 respectively. The maximal requested volume is 4. The required valuation is $10 * (x_1 + x_2) + x_1 + 2x_2$, assuming that $x_1 + x_2 \leq 4$.

In \mathcal{L}_{CB} and \mathcal{L}_{CG} the most natural way is to use OR operator, e.g.: $\langle\langle c_1, 11 \rangle \text{ OR } \langle c_2, 12 \rangle\rangle$.

Again, the \mathcal{L}_{CGB} seems to be the most intuitive, since it directly reflects the structure of the utility function: $\langle\langle(1, 0), 1, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4\}\rangle \vee \langle(0, 1), 2, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_2 \leq 4\}\rangle, 10\rangle$.

6 Summary

We have introduced three families of bidding languages for continuous auctions. They are based on the concepts derived from the well-defined combinatorial auctions. We have generalized the language families based on goods, bids, and on both of them to the continuous case. We have also generalized several notions, which creates a solid ground for bidding languages in a continuous case. Some of the properties known from their equivalents in combinatorial auctions are preserved in the proposed languages. Almost all introduced languages are fully expressive, but they differ in succinctness and logical grounds from the agent point of view. No language is dominating in the meaning of these criteria. Further work should include deeper analysis of the languages in a context of their expressiveness and succinctness for particular classes of valuation functions.

Acknowledgements. The research was supported by the Polish National Budget Funds 2010-2013 for science under the grant N N514 044438.

References

1. Benyoucef, M., Pringadi, R.: A BPEL Based Implementation of Online Auctions. In: Georgakopoulos, D., Ritter, N., Benatallah, B., Zirpins, C., Feuerlicht, G., Schoenherr, M., Motahari-Nezhad, H.R. (eds.) ICSOC 2006. LNCS, vol. 4652, pp. 104–115. Springer, Heidelberg (2007)
2. Boutilier, C., Hoos, H.H.: Bidding languages for combinatorial auctions. In: Proc. 17th Intl. Joint Conference on Artif. Intell. (2001)
3. Hoos, H.H., Boutilier, C.: Solving combinatorial auctions using stochastic local search. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence, pp. 22–29 (2000)
4. Kaleta, M., Traczyk, T. (eds.): Modeling Multi-commodity Trade. AISC, vol. 121. Springer, Heidelberg (2012)
5. Lehmann, D., Müller, R., Sandholm, T.: The winner determination problem. In: Cramton, P., Shoham, Y., Richard, S. (eds.) Combinatorial Auctions, ch. 12. MIT Press (2006)
6. Nisan, N.: Bidding and allocation in combinatorial auctions. In: Proceedings of ACM Conference on Electronic Commerce, pp. 1–12 (2000)
7. Rolli, D., Eberhart, A.: An auction reference model for describing and running auctions. In: Proc. of the Wirtschaftsinformatik (2005)

Auction of Time as a Tool for Solving Multiagent Scheduling Problems

Piotr Modliński

Abstract. The paper presents an idea of multi-agent scheduling problem as an extension of the classical models. In contrary to the classical approach, where the individual operations have pre-defined weights, a lot of players (agents) with different (often conflicting) preferences for performing various operations appear. A formal model of the problem is formulated and the general concept of scheduling problem formulation in the market form is presented. In the paper the exemplary application of the market mechanism for the specific problem appears.

1 Introduction

The classical approach of ranking and scheduling is based on an assumption that the collection of processors (homogeneous or heterogeneous) should execute the set of tasks in order to obtain specified goal. The individual models differ in a goal specification (for example minimization of medium/maximum late/delay, maximization of the number of completed tasks, etc.), set of machines (homogeneous ones, varying in efficiency or functionality) and definition of tasks (divisible or indivisible ones).

It is assumed that in each of the cases we have the determined set of essential parameters of the tasks as well as the processors, that are well-known to every participant. The queuing theory considers specific issue of operating systems, where the distributions of the specific task in a given moment, and the completion time are given. This approach has long been used in analysis of mass service in telecommunication system.

As different projects, e.g. IT, get more and more complex, their appropriate management becomes a very important issue. Although all essential resources need to

Piotr Modliński
Institute of Control and Computation Engineering,
Nowowiejska 15/19 00-665 Warsaw, Poland
e-mail: p.modlinski@ia.pw.edu.pl

be taken into consideration, the time seems to be the crucial one. As the time could not be stored or stopped, and every single project has more or less rigorously determined deadline, taking the time into consideration becomes necessary during the project management. It is possible (at least in theory) to determine when and which tasks should be completed. However, when we deal with people not with machines, their time preferences become essential due to the efficiency of the workers, correlated for sure with their satisfaction [5]. A different problem, which is rather within ethical area, is constraint to unsatisfactory work [4]. Therefore, it is difficult (or even impossible) to state in general which time-limits are better or worse.

People should have possibility to express their individual opinions. From a formal point of view the problem of scheduling of university classes is quite similar [3].

2 Problem Formulation

Let us consider the problem of finding a schedule from the point of view of independent agents. We have to include availability of resources (also agents) as constraints (one resource can't be used in the same time in different places) and as preferences (each agent may prefer some periods). If in addition we assume, that every agent may want to be included in some jobs, the problem becomes complex.

Let us take the set \mathcal{T} of disjoint and identical¹ periods $t \in \mathcal{T}$, and the set \mathcal{R} of single indivisible resources r . Every agent $a \in \mathcal{A} \subset \mathcal{R}$ can be seen as a resource, that may be needed to finish some operation. Every resource may be available or unavailable in the specified period. It is defined by a matrix \mathbf{D} , where single elements d_{rt} are equal to 1 if resource r is available in period t , or 0 otherwise. Resources can be gathered in classes ($c \in \mathcal{C}$). In that way, the class can be considered as subset of set of resources ($c \subseteq \mathcal{R}$). Every resource can be a part of many classes, and each class can include any number of resources (but empty class is probably a bad idea).

Assume, that an objective is a completion of the jobs from a set ($z \in \mathcal{Z}$). Every job is described by a set O_z of indivisible operations, each lasting one period. Each job has a weight – the higher the weight, the more important the job is. Resource requirements are defined at the level of operations, so that different resources can be used in various operations within the same job². The individual operations are not ordered. The purpose of the problem is to maximize the weighted number of accepted jobs. Each job can be accepted iff all included operations are accepted. That problem can be easily formulated as the mixed integer, or combinatorial programming task.

The situation changes when the agents are treated as *autonomous entities* with their own preferences for tasks in which they wish to participate, as well as the corresponding terms. We need to formulate a new problem. We cannot just

¹ From model's point of view – every period has identical length – every operation may be done in every period if other constraints allow this.

² The only exception are the requirements for agents, but they are the result of their autonomy, as described later.

maximize single objective function, but rather need to use multi-criteria optimization. Each solution (allocation) is no longer evaluated in centralized manner (as in single objective function), but independently by each agent. Our aim is to maximize satisfactions of all agents. It is easy to define rules to find Pareto-optimal solutions, but it is not trivial to select the one, which is the best. In general we cannot even unambiguously define what does in mean "best". The difficulty is greater because real preferences are private informations of players, and they may not want to disclose these.

3 Time Auction

The problem discussed in section 2 can be formulated as a market mechanism, where the exchanged goods will be times of availability of various resources. Such mechanism called "Time auction", or "Auction of time" was proposed by the author in the paper [3], and is schematically shown in figure 1.

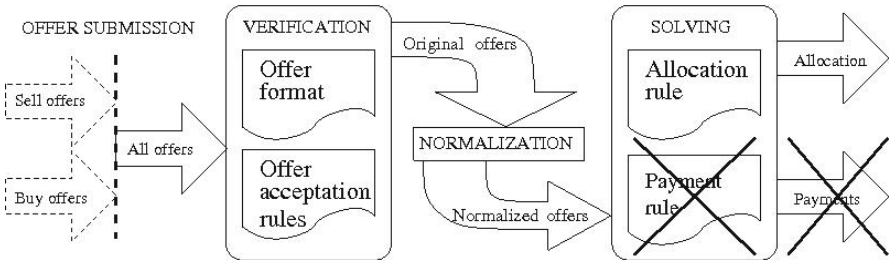


Fig. 1 Schema of time auction

Having set their own preferences, agents introduce offers for purchasing and selling specific goods. In presented mechanism the multi-commodity offers for bundles of goods, called *bundled offers* [1], are very important. After verification of correctness³ and the formal acceptability⁴ only offers, that potentially could be accepted, remain. Next step is the normalization – recalculating of all offer prices into a one ratio scale. This is necessary because of the difficulties in ensuring incentive compatibility [7] – every player has own preferences. From a global point of view the solution will be effective if the offer prices reported by players are true. Given the fact that the auction result is only the allocation of time – there is no information about the payments⁵, the players have no incentive to disclose their preferences. The

³ Proposed format is OpenM3 – see [1, 2], but proving it’s usefulness for writing this type of problems is beyond the scope of this paper.

⁴ i.e. if the player has the right to make such an offer basing on external rules.

⁵ Because it is not a tool of market exchange, but only to facilitate the negotiations.

proposed normalization is a generalization of the method described by Strevell and Chong [6]. The last step follows when the prices of all bids are normalized so that where possible reflect the level of satisfaction of each player from the acceptance of each offer, or the whole solution. Since this is a common auction problem, almost every mechanism able to solve such problem should be equivalent to finding the optimal schedule.

3.1 Goods and Offers

As mentioned earlier, goods used in the time auction are periods of availability of the resources. In this way they are defined by the matrix \mathbf{D} (see [2]) – we can use only these goods (pairs (r, t)), for which $d_{rt} = 1$. Of course they must be placed on the market – the balance requires that someone can buy the good only if it is sold by somebody. For this purpose the additional (artificial) zero-price offers exist. The artificial offers bring goods (moments of resource availability) to the market. These offers may be submitted either by the market operator, or by the player actually managing the resource. In the same way each player adds offers specifying their availability (as said before, each agent can be understood as a specific resource). In other words, each player a needs to fill a -th row of matrix \mathbf{D} . Additionally we can define "buy" offer for places in specific job. With that mechanism we are able to express our interest in a specific job. All of the above offers are *simple offers* as classified in [1]. In the model described we also have *bundled offers* [1], which can be used to exchange on goods for others, in particular to implement classes (virtual resources), and jobs consisting of operations that require specific resources.

Note, that it is needed to reject signals that are invalid for various reasons. In particular, the terms of the offer, to make that a player has no possibility, for example, tries to provide resources, which in fact he does not have. In this paper, this point will not be further discussed. We will assume, that the offers submitted are correct.

3.2 Normalization and Offer Prices

Suppose, that each player a is able to define for each of its offer j_a real utility $U_a^*(j) \geq 0$ in any ratio scale. We cannot assume much about these utilities because neither the offer, nor the generated schedule, have a direct impact on material or financial values. On the other hand, every agent has to pass some numerical signals, defining that utility to the balancing mechanism. Due to the information privateness, agents may behave strategically. The signal presented by the player is the modified utility $U_a(j) \geq 0$, where $Z_a(j)$ is the modification of each offer (j_a) – see equation [1]. In that context, the purpose of normalization is to bring utilities of all players into the same scale, and to minimize modifications of utilities.

$$U_a(j) = U_a^*(j) + Z_a(j) \quad (1)$$

We can interpret utility $U_a(j)$ as a satisfaction of the player a for acceptance of an offer j . Our considerations can be generalized to the case of partially accepted offers. Assuming a linear form of the utility function, we can calculate satisfaction for an offer accepted in part $s_j \in [0, 1]$ as $U_a(j) \cdot s_j$. It can be shown, that by increasing the value of $Z_a(j)$ the probability of accepting offer j increases. In this way, a rational player would introduce positive modifications. To prevent such behaviours we introduce normalization based on utility of solution. *Solution of time auction* is the vector \mathbf{s} , which elements $s_j \in [0, 1]$ define offers acceptance⁶. Then, for linear utility function, we can calculate utility of solution (from player's point of view) as the sum of utilities of accepted offers⁷ – see equation (2). Similarly, we can find $U_a^*(\mathbf{s})$ for private values.

$$U_a(\mathbf{s}) = \sum_{j \in \mathcal{J}_a} U_a(j) s_j \quad (2)$$

Let introduce for every player so called *optimal pseudo solution* \tilde{s}_a as total acceptance of all offers made by the player. We can calculate utility of such solution without worrying about the offers submitted by the other players (see (3), (4)).

$$U_a(\tilde{s}_a) = \sum_{j \in \mathcal{J}_a} U_a(j) \quad (3)$$

$$U_a^*(\tilde{s}_a) = \sum_{j \in \mathcal{J}_a} U_a^*(j) \quad (4)$$

After calculating utilities, we can determine normalized offer prices, which will always be in the range $[0, 1]$ by dividing utility by the calculated value (see eq. (5)).

$$P(j) = \frac{U_a(j)}{U_a(\tilde{s}_a)} \quad (5)$$

3.3 Properties of Normalized Prices

In the case when player introduces proportional disturbing into reported offers (in other words, for every $j \in \mathcal{J}_a$ we have $Z_a(j) = z_a \cdot U_a^*(j)$) it can be shown that it cannot change offer prices in any way. It is worth noting that this disorder is equivalent to presenting utility on a different ratio scale.

Much more interesting is the case in which the agent introduces any of disturbing. Agent can indeed improve the competitiveness of selected offers, but does so at the cost of remaining ones. Summarized utility of all player's offers will always

⁶ From $s_j = 0$ for offer j rejected, to $s_j = 1$ for accepted.

⁷ To simplify the notation in equation (2) only offers of the player J_a are summed, because utilities of other agents are equal to 0.

be constant and equal 1. Usefulness of such mechanism with constant amount of resources which could be assigned was proved by Strevell and Chong [6]. In general, the time auction in which there is a number of groups of offers (i.e. offers specifying times, another of jobs, etc.) normalization should be carried out separately for each group.

3.4 Solution

The set of offers and goods presented above together with the allocation rule is a market problem (see fig. 1). The example of such allocation rule and mechanism able to solve such problem is presented in [3]. Due to lack of determination of settlement prices, much of the problems that may occur is irrelevant. The algorithm for solving allocation problem will not be further discussed in this paper.

4 Practical Examples – Production Line

As it was mentioned in the introduction, the presented mechanism has a potential to improve the performance and satisfaction of the work by increasing the impact of preferences on the performed tasks and the working time. In the considered examples, we concentrate on a simple task of assigning employees to the production line, however various methods of defining preferences will be shown. What is important, every agent can use any of these methods no matter what the others do. Presented examples are not intended to exhaust the issues related to production management, but to show a possibility of practical application of the time auction, so there is no comparison with other solutions used to solve similar problems.

In every example we consider one day (three 4-hour shifts) of production line. We have 8 employees divided into two groups – 2 of them (a_1, a_2) are "managers", 6 of them ($a_3 - a_8$) are "workers". In every moment we need one manager and 3 workers. Another requirement is that an employee can work only 8 hours a day. It is not defined, when jobs are being done – schedule is a result of the optimization.

The first step for every employee (agent) is to define, which moment is available for him. Assume, that the term t_1 is unacceptable for an agent a_7 , t_2 for a_2 and t_3 for a_4 . All the others terms are acceptable for all agents. Every agent has to prepare and submit his selling offers for particular time slots⁸. Every employee puts on the market his own time in convenient terms. Offers are presented in table 1.

The next step is to check qualifications, and it can be done automatically. Basing on employee data, the mechanism can define the set of bundle offers, each to exchange the time of employee and virtual good – time of "manager", or "worker" shown in table 2.

⁸ That means, everyone except a_2, a_4 and a_7 submits three offers, and those three employees only two offers.

Table 1 Selling employee’s time offers

Good	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Agent 8
(T_1, a_i)	-1	-1	-1	-1	-1	-1		-1
(T_2, a_i)		-1		-1	-1	-1	-1	-1
(T_3, a_i)			-1			-1	-1	-1

^a Every agent bids on different goods. Table above is simplified for the sake of limited size of the paper. In details the structure of the offers is described in work [3]
^b empty fields are zeros

Table 2 Exemplary offers changing time of employees (a_1 and a_3) for virtual goods

Good (Time, Resource)	Offers of a_1	Offers of a_3
(T_1, a_1)	1	
(T_2, a_1)		1
(T_3, a_1)		
...		
(T_1, a_3)		1
(T_2, a_3)		
(T_3, a_3)		1
(T_1, M^a)	-1	
(T_2, M)		-1
(T_3, M)		
...		
(T_1, W^b)		-1
(T_2, W)		
(T_3, W)		-1

^a M – virtual resource "Manager"
^b W – virtual resource "physical Worker"

Resource requirements at different times are also described by bundled offers. For every job⁹ (Z_1, Z_2, Z_3) we need one manager and three workers. Offers are presented in table 3. Offer prices are very important (even if all of them are equal), otherwise, if the offer prices are equal to zero, all offers will be rejected. The natural solution is to determine the benefits of the planned work of the shift.

Auction described in such way does not take into account two elements. First – employee can not work more, than for two terms. Second – we cannot describe preferences about terms. We are only able to mark some terms as available, and other as unavailable. We will solve these problems together.

Employees can solve this in at least two ways – simpler, which does not take into account dependencies between terms (that means defining preferences for each term separately), and more complicated, which allows showing every convenient solution (for example we can request continuous time – we won’t accept solution $\{t_1, t_3\}$).

⁹ Each job needs working for 4 hours.

Table 3 Offers describing requirements of each job

Good (Time, Virt. resource)	Job Z_1	Job Z_2	Job Z_3
(T_1, M)	1		
(T_2, M)		1	
(T_3, M)			1
(T_1, W)	3		
(T_2, W)		3	
(T_3, W)			3
Offer price P_{Z_i}	10	10	10

In both methods we have to define an artificial good for each employee, but its interpretation will be different.

4.1 Interdependent Terms

Let us define for each employee a artificial commodity independent of time (s_a^1), which will determine the number of changes, in which he can work. We will insert it to the market with additional simple offers submitted by each agent, or by the agent who represents the employer. Since each of employees has a predetermined maximum dimension of work, it can be done automatically. Sample offers are presented in table 4.

Table 4 Offers adding artificial commodities of the 1st type

Commodity	Offer of a_1	Offer of a_2	Offer of a_3	...
Artificial of 1st agent ($s_{a_1}^1$)	-2			
Artificial of 2nd agent ($s_{a_2}^1$)		-2		
Artificial of 3rd agent ($s_{a_3}^1$)			-2	...
⋮				
Offer price ($P(of)$)	0	0	0	...

To make use of these products it is necessary to replace simple offers shown in table 3 by bundled offers changing piece of artificial commodity with specific availability time of agent. Examples for agents a_1 and a_2 are shown in table 5 below.

Since there are only two commodities for each player, there is no possibility that he adopts more than two terms. This solution, however, carries the risk that the employee will be forced to work on the first and third shift, and this may be

Table 5 Example of using 1st type artificial commodities

Commodity	Offers of 1st agent			Offers of 2nd agent		...
(T_1, a_1)	-1					
(T_2, a_1)		-1				...
(T_3, a_1)			-1			
(T_1, a_2)				-1		
(T_2, a_2)						...
(T_3, a_2)					-1	
⋮						⋮
Artificial of 1st agent ($s_{a_1}^1$)	1	1	1			...
Artificial of 2nd agent ($s_{a_2}^1$)				1	1	

unacceptable to him. This problem can be solved by using artificial commodities of 2nd type, as shown in section 4.2

4.2 Dependencies between Shifts

In the previously considered cases, the player described terms of his availability. Instead he can specify acceptable sets of terms. Again, consider the example of a player a_1 . In theory, all three terms fits, but he may put offers not for sale individual terms, but specific groups¹⁰. We have six possible configurations of meeting the working time limit ($\{T_1\}$, $\{T_2\}$, $\{T_3\}$, $\{T_1, T_2\}$, $\{T_1, T_3\}$, and $\{T_2, T_3\}$). Assume that the employee is not satisfied to spend at work one shift as idle waiting – the set $\{T_1, T_3\}$ is unacceptable. In that case, instead of offering single periods, player bids for the whole packages (without the unacceptable one), as shown in table 6

Table 6 Example of the usage the 2nd type artificial commodities and description of acceptable sets of terms

Commodity	Offers of player a_1					...
(T_1, a_1)	-1			-1		
(T_2, a_1)		-1		-1	-1	...
(T_3, a_1)			-1		-1	
2nd type artificial ($s_{a_1}^2$)	1	1	1	1	1	...
1st type artificial ($s_{a_1}^1$)	1	1	1	2	2	...

¹⁰ Perhaps one-element.

Commodity $s_{a_1}^2$ is introduced to the market similarly to the described in previous section $s_{a_1}^1$, but it is a indivisible commodity (volume $\in \{0, 1\}$). By that way we have to exclude the offers. As shown in table 6, artificial commodities of the first and second type can be used independently of each other. On one hand, this enforces players to control allowable working times, and on the other gives flexibility in the description of their preferences. Preferences of specific terms, or whole sets can be expressed by offer prices (which are normalized as presented in section 3.2), but these prices are treated as sale prices of player's time, so should be higher, if the player less prefer specific term (to define the level of dissatisfaction).

5 Summary

The paper presents the idea of the time auction mechanism, and shows a practical example of its application to the planning work including the preferences of individual employees. In presented examples, no additional resources were included, which in a real system would be used because of its transparency, but could be treated in a manner similar to agents who have no preference – offers submitted for each shift of zero rates allow for the introduction of resources, “on the market”, and then to use them for specific tasks.

Acknowledgements. The research was supported by the Polish National Budget Funds 2010-2013 for science under the grant N N514 044438.

References

1. Kacprzak, P.H., Kaleta, M., Pałka, P., Smolira, K., Toczyłowski, E., Traczyk, T.: M3: Open multi-commodity market data model for network systems. In: 16th International Conference on Systems Science, pp. 309–319 (2007)
2. Kaleta, M., Traczyk, T. (eds.): Modeling Multi-commodity Trade. AISC, vol. 121. Springer, Heidelberg (2012)
3. Modliński, P.: Problem harmonogramowania jako kombinatoryczna aukcja czasu. Studia i Materiały Polskiego Stowarzyszenia Zarządzania Wiedzą 33 (2010) (in Polish)
4. Paul, J., Catholic Church. Pope (1978-2005 : John Paul II), Gałkowski, J.: Laborem exercens: tekst i komentarze. Jan Paweł II naucza. Red. Wydawnictw Katolickiego Uniwersytetu Lubelskiego (1986) (in Polish)
5. Schultz, D.P., Schultz, S.E., Kranas, G.: Psychologia a wyzwania dzisiejszej pracy. Wydawnictwo Naukowe PWN (2006) (in Polish)
6. Strevell, M., Chong, P.: Gambling on vacation. Interfaces 15(2), 63–67 (1985)
7. Toczyłowski, E.: Optimization of Market Processes under Constraints. II extended edition. EXIT Academic Publishing (2003) (in Polish)

Application of an Auction Algorithm in an Agent-Based Power Balancing System

Piotr Pałka, Weronika Radziszewska, and Zbigniew Nahorski

Abstract. The paper presents an application of an auction algorithm in a computer multi-agent system for managing the unbalanced energy in a microgrid. The main goal of the system is to control and minimize the differences between the current energy demand and the actual energy production, using an auction algorithm. The assumption of distributed generations in the microgrid, which includes renewable power sources, is made. The storages, and the controllable power sources improve the system operation. The differences between the actual demand and generated energy are caused by unpredictable level of electric power generation by uncontrolled sources (mainly wind turbines and solar panels) and/or randomness of power utilization. The system will tend to balance these differences on-line in short time intervals (about one minute) to follow-up the varying level of local power generation and loads.

1 Introduction

The renewable energy sources develop rapidly over recent years. The idea of dispersing the sources, mainly renewable ones, within the power grid is very promising. This is essentially connected with the prosumer concept [20]. A prosumer is an entity that not only purchases energy, but can also produce and export it to the power grid. With such configuration the need for new, efficient, and reliable management systems appears.

Piotr Pałka

Warsaw University of Technology, Institute of Control and Computation Engineering,
Warsaw, Poland

e-mail: P.Palka@ia.pw.edu.pl

Weronika Radziszewska · Zbigniew Nahorski

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

e-mail: {Weronika.Radziszewska,Zbigniew.Nahorski}@ibspan.waw.pl

Traditional energy management systems with centralized structure fail to provide well-suited solution to recent concepts. This is caused mainly by the traditional system assumption of unidirectional flow of energy, from the distribution companies to the loads, located in the leaves of the distribution grid. Distributed generation inside the distributed grid ruins this assumption, as the energy flows bidirectionally, and need for a new management systems appears. As generators are dispersed in the grid, the idea of decentralized management system can be very promising. Recently, decentralization of decisions in computer networks is realized more and more often by multi-agent systems [17]. This solution is also applied in the considered energy management system. Agents are connected with devices, like power sources, loads, and storages. They have private knowledge and individual goals defined. Agents communicate with others in order to ensure the security of energy supply, and to reduce (minimize) the unplanned shortages or surpluses. Thus, both sides, the supply and the load devices, take part in management of the energy. This forms a distributed energy management system. Decentralized decision making systems, implemented as the multi-agent systems, has been already described in the literature. Authors of [8, 9] describe multi-agent systems applied in the energy issues, basing on previously published papers [1, 7, 14]. The control and minimization of the differences between the current energy demand and the long-time plan is one of main issues of Agent-based Power Balancing System for the Microgrids [12]. The developed multi-agent system assumes balancing the differences in short time intervals. The differences are caused by unpredictable level of dispersed, renewable sources of energy, and by variations of the actual demand. An auction is a well-suited solution to solve the problem with decentralized, autonomous parties that tend to realize only its own goals. As in the actual trading, particular entities can reach sub-optimal goods allocation in the competitive environment, even without the assumption of shared knowledge. Thus, in the Agent-based Power Balancing System for the Microgrids the trading of the unbalanced energy is performed to minimize differences between actual energy production and consumption. As short reaction time as possible is looked for to suppress imbalance, and to lower the costs borne by devices owner. The goal of the paper is to present the auction algorithms applied in the system.

2 Related Works

The idea of a prosumer power grid, which seeks to balance generation and power demand, is introduced in [20]. Due to dynamic generation and demand of electric power, and the need to obtain the power balance, it is required to apply more complex control system than classical automatic regulation used at present. The literature suggests that the multi-agent systems may be a promising solution of this problem.

Management of the power distribution in the power grids develops rapidly. Recent concept of supplying the loads in the distribution grids differs from the previously used structures, in which the distribution grid is supplied from the high

voltage power grid. The new concept of smart-grids – the subgrids with bidirectional power and information flows, are currently being considered. The efficiency of these sub-grids depends mainly on the power balancing technique. For majority of power generators existing in the sub-grid, like wind turbines or solar panels, the level of produced power depends strongly on the meteorological conditions. They have no automatic mechanisms to self-adapt energy level production to existing demand, such as are used in the large power stations. Thus the level of energy produced in dispersed generation microgrids is to a large extent random. Moreover, due to relatively small number of loads and generators in a microgrid, there is no strong averaging as in large grids, and the energy consumption is characterized by high volatility. This fact significantly hinders forecasting the demand for power consumption. Both these factors put considerable demands on the management system and balancing power flows in prosumer sub-grids.

In order to keep the power integrity in the small power grids, it is often necessary to apply complex energy management systems (EMS). These systems often comprises the control subsystems, oriented on optimization of the grid operating costs, cooperation with the distribution grid operator, and reliability of the supply of the energy. Another goal of the system can be load balancing, load reduction, acquiring additional supply of the energy in the peak, or increasing the load during the off-peak periods [1, 13, 19]. Particular interesting are EMS systems designed for distributed energy management, and those in which auction is used. Paper [5] deals with auction in the distribution grid with the microgrids, including those with the dispersed generations. Each generator willing to sell the energy declares the prices for every consumer. The price can differ for different consumers. The resulting auction price, that is used to associate the generators and loads, is not the transaction price in the strict sense. Despite using the auction terminology, the auction is only a method for the pairs association, in order to maximize the overall accepted offer prices. In turn, the paper [14] describes the multi-agent system application for a public facility, powered from the distribution grid, with installed distributed generators. The system comprises of agents representing dispersed generators, storages, and loads of the energy. The goal of the system is to ensure power balance, and minimization of the cost of the purchased energy from the distribution grid. The paper [7] comprises more details of the multi-agent system structure.

The distributed management systems uses so called auction algorithms. The auction algorithm idea is to solve a decision or computation problem by multiple autonomous actors. The idea has been considered for a long time. The author of [4] suggests application of the auction theory for solving linear programming models (scheduling problems, matching problems). Similar idea is the Contract Net Protocol proposed in the paper [18]. It is a meta protocol, which can be used to solve decision or computation problems by tasks delegation. The tasks are delegated to interested entities, and allocation of tasks to entities is based on an auction process. Entities that are interested in cooperation make bids, and the task delegating entity acts as the auctioneer that choses most profitable bids, delegates tasks, and finally awards entities for task execution. Similar solution is applied in the Auction-Based Routing Algorithm (ABRA) for the delay and disruptive tolerant networks [15].

ABRA assumes that autonomous mobile nodes perform auctions in order to establish a route for data packets. However, negotiations can also be used for solution of the complex issue. The monotonic concession protocol for the multilateral negotiations is adopted in [21], where authors consider negotiation to solve a production sequencing problem in the car factory. Thus the auction or negotiation algorithms are very promising tools for solving complex optimization problems with incomplete knowledge.

3 Agent-Based Power Balancing System

Each generation unit, including renewable dispersed sources, traditional generators, groups of possibly aggregated loads, and energy storing devices, is represented by a corresponding group of autonomic software components. These components interact each other to reduce imbalance. According to the agent-based programming paradigm [16], the software components can be treated as autonomous agents. Each agent has been defined its own goal. The goal is modeled by different roles for individual agents, while the common aim is to ensure suitable working conditions for particular loads, and to pay less for the unbalanced energy, which is much more expensive than the contracted one. But each agent can have different goals, and in order to meet them, agents interact mutually. Above assumptions cause that the system satisfies the multi-agent system conditions [17]. The overall goal of the system considered is to ensure security of supply and imbalance reduction.

As it has been noticed in [12], for the sake of the system design, it is worth to divide devices into two groups. The first group comprises of devices that can control the level of produced or consumed energy, provided appropriate technical constraints are met. The second group cannot control it, as the level of consumption or production depends on meteorological or hydrological conditions, or finally on the unpredictable human behavior. Note, that each group can contain both generators and loads. The first group is called **controllable**, and contains a reciprocating engine, thermal units, micro cogeneration units, and hydro turbines. Also some of loads, that are able to control its demand, like smart refrigerators, can be included into this group. The second group is called **uncontrollable**, and includes majority of loads, excluding those, which are able to control its demand, and also cooperate with the balancing system, and majority of renewable sources, mostly wind turbines and solar units. Storage devices are included in both groups, and act either as the controllable when discharging, or as the uncontrollable when charging.

The Agent-based Power Balancing System for the Microgrids is implemented using JADE 4.0 framework [3], and the Java 1.6 language. Eight main agent roles are designed and implemented: the active and passive **modeler** that models the physical behavior of the devices in a power generation or consumption states, taking into account the meteorological conditions; the active and passive **predictor**, which provides short-term forecasts of demand or energy production, taking into account the meteorological conditions; the active and passive **negotiator** that negotiates the

delivery of deficiency or the reduction of energy surplus. Three above listed agent types, are implemented for the controllable ('passive' ones) and for the uncontrollable ('active' ones) devices, but the implementation details differ. Besides, the **morris column** agent, and the **external grid agent** are implemented. In addition, the **monitor** agent can be considered, which goal is to monitor the state of particular agents. Detailed description of particular agents can be found in [10, 11, 12].

An active modeler agent periodically reports informations about regulatory capabilities of the device. The regulatory capabilities can be considered as permissible increase or decrease of produced or consumed energy from the device working point. The working point can be positive (when device generates energy), zero, or negative (e.g. when energy storage unit is charged, or if the device consumes the energy). The active modeler compares the actual working point with the planned one, and with the contracted energy. It receives the short-time forecast from predictor agent, and on the basis of a mathematical model of the device, determines the working point in the future, with predetermined (short-time) horizon. In result, it may conclude that it entered, or may enter into the imbalance state.

When the imbalance state is detected, the active modeler requests the active negotiator to reduce the imbalance. The process of the imbalance reduction using the auction algorithm is described in more details in section 4. A passive modeler agent also checks the current working point, and publishes its regulatory capabilities at the morris column agent. Moreover, it obtains the forecast from the passive predictor, and publishes the future working points. Both active and passive predictor agents provide the short-time forecasts on the basis of the weather forecast, time of day, day of week, and season. Morris column agent acts as the public repository. It provides other agent with the possibility to publish, look for, remove, and update information about its actual and predicted regulation capabilities. The external grid agent trades with the external grid. It is active only when the microgrid is connected to an external active grid.

4 Auction Algorithm for Energy Balancing

In the paper we omit the issues of communication among the agents, and the description of elements of the system, as these are the topics of another publications. The object of the trade is the actual or predicted lack or excess of energy. Note, however, that time structure cannot be neglected. Each imbalance is characterized by its size, and by the moment of time, when the imbalance is detected or predicted. Thus, the multi-commodity trade is performed, where the main focus has been placed on the real-time trade of commodities differentiated by the moment of realization.

When the appropriate agent (the active modeler agent) detects the imbalance (which can be actual or predicted), the main negotiation process begins. Note, that each new negotiation process runs parallel to the already existing ones. Moreover, the particular negotiation processes are isolated. Each imbalance causes the appropriate auction algorithm execution, which goal is to eliminate, or at least minimize

the imbalance. The trade process should be very quick, to achieve fast imbalance reduction. Multiple instances of negotiation takes place at the same time, so it is important to ensure that the individual negotiation are processed reliably, without interfering each other. To act swiftly, the auction algorithm should be very simple. The negotiation method described in this paper is chosen to be an one-side, sealed bid auction. The auctioneer can sell the excess of energy, or purchase it when it is lacking. Each active negotiator can initiate the trade, thus there does not exists single, centralized entity that manages the trade. Actually, ad-hoc auctions are executed, operated by active negotiators.

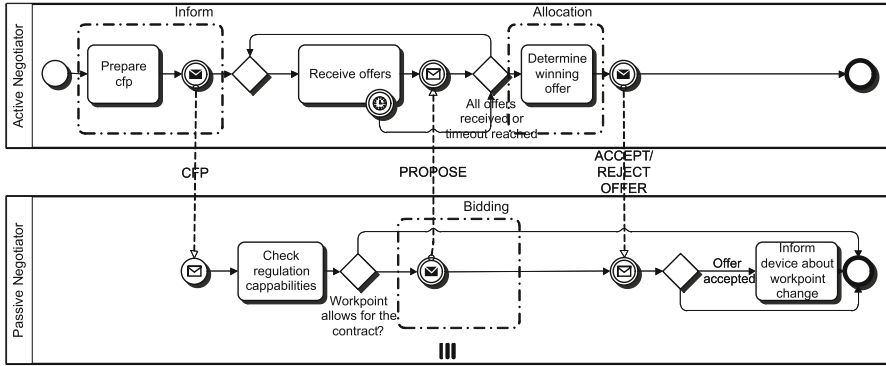


Fig. 1 Single auction process presented on the BPMN collaboration diagram

In Fig. 1 the single auction process is presented. BPMN 2.0 notation [2] is used due to its convenience and transparency. It provides not only the flow of communicates, as in the UML or AUML, but also the inner processes of particular agents, and their decisions, associated with sending or receiving of respective communicates. The active negotiator initializes the auction, by sending the Call For Proposal (CFP) communicate to passive negotiators, that have been preselected as suitable entities for imbalance reduction. The active negotiator, which initiated the auction, waits for the offers for specified time (e.g. 100 ms). For a passive negotiator, the auction process begins in the moment of obtaining the CPF message. In this way we model the situation, where each new CFP communicate received causes the new parallel auction process. The passive negotiator checks, if its actual workpoint and its production bounds allow for imbalance reduction. If the imbalance is positive, i.e. the device that is represented by the auctioneer deals with the excess of energy, the device that is willing to reduce it should decrease its actual working point by the imbalance value (to the accuracy of grid losses). Similarly, if the imbalance is negative, i.e. it causes the lack of energy, the device that is willing to reduce it, should increase the working point, viz. produce more or consume less. Note, that the devices reducing the imbalance are not only generators, but also the energy storages or controllable loads.

If an agent determines that it is able to deal with imbalance, it submits an offer (PROPOSE) to the active negotiator, and waits for an answer. When the active negotiator collects all offers, or if the timeout is reached, it goes to the allocation phase, in which it decides which offer to choose. The decision is based on the allocation rule. In the system, the allocation rule is the sealed-bid auction allocation rule, so the most profitable offer is chosen. When no offers are submitted, it means that the active negotiators cannot deal with the imbalance. However, if the exchange with an active external power grid is possible, such situation cannot occur. When the offers are allocated, and the 'winning one' is chosen, active negotiator sends to each of the passive negotiators that send an offer, either the communicate ACCEPT_PROPOSAL when the agent submitted winning offer, or REJECT_PROPOSAL otherwise. When the passive negotiator obtains ACCEPT_PROPOSAL message it informs its modeler agent, and device agent to change the working point. At the same time, the active negotiator informs its device modeler agent about satisfying the demand.

5 Preliminary Results

Auction processes durations, from noticing the imbalance to its reduction (by agreement between the agents) are presented in Fig. 2. Test was here executed for an extended period of over 27 minutes. The imbalance states last very shortly, by average the 48,43 ms, with the median of 36 ms. At some points, the time required to reach the balanced state reached 1 second. This is a rather undesirable long time, which is assumed to be connected to the strong multi-threading of the system (each device uses at least three threads) and also to the problem with concurrent access to the resources like databases. An important conclusion is that the balancing can be done in less than two seconds for this ten-device system and this number is not much different from the cases with three or five devices. This means that the response time of the system between detecting the imbalance and suppressing it is so short that it can be considered as almost a real time one.

In the first case study, the inability to balance the energy is due to too small amount of energy consumed in comparison to the generation. The test includes two 50 kWh engines, but they cannot work below the minimal working level which is 12,5 kWh, unless they are turned off. Engine 1 is slowly lowering it's working point to this level. Due to the island operation mode assumed and no storage device, the energy generated by the wind turbine produced is wasted. The aggregated energy in the whole system is presented in Fig. 3. The generations on the one side and the loads on the other are aggregated to one second intervals to demonstrate the relation between supply and demand. The test lasted 71 seconds, but first few seconds, that included initialization of agents, were omitted. Some small differences between production and consumptions are visible: this is due to lag time in balancing, and also due to nine imbalanced energy cases.

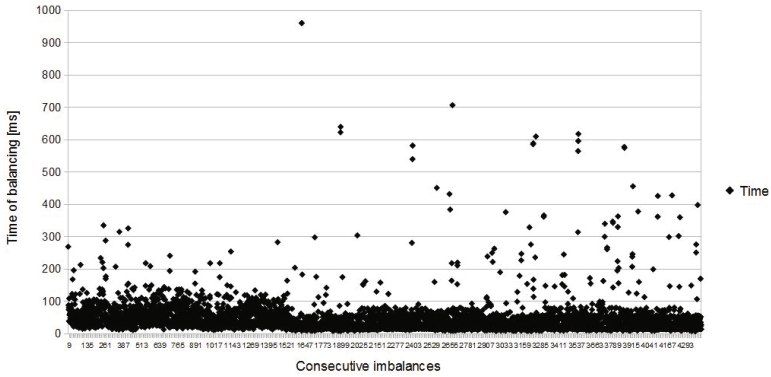


Fig. 2 The times of imbalance reduction

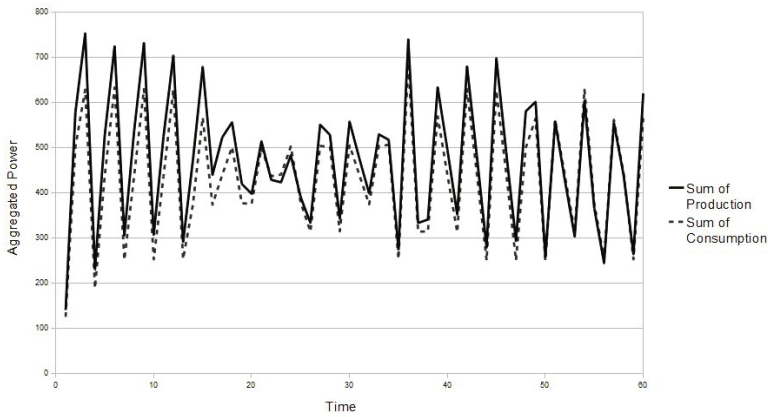


Fig. 3 Aggregation of production and consumption of energy during one second of the run of test case 1

In the second case study, one of the engines was replaced with an energy storage device (battery). Energy storage offers its energy to the consumers at the price depending on the current charge level of the battery. That gives agents an incentive to use battery only if it is sufficiently charged. Battery is actively trying to charge while it is approaching the discharged state. In this test case the produced power and the consumed power were more closer each other, and the engine had to increase its working level while the battery was in a discharging trend. There were no unbalanced states detected during the whole test. Fig. 4 presents the energy variation during the second case study execution.

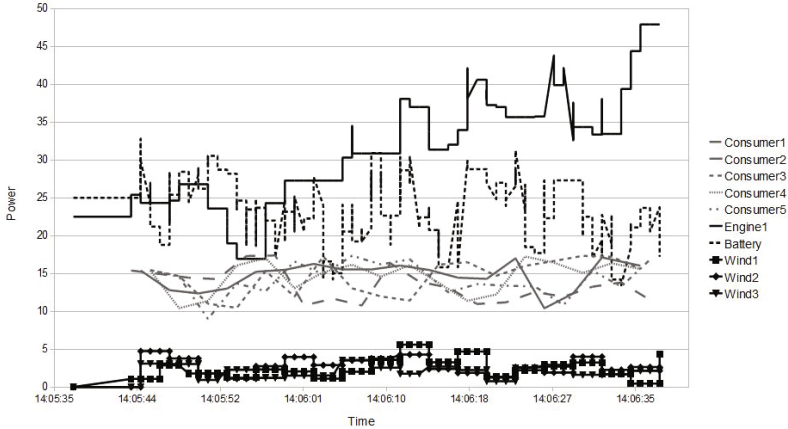


Fig. 4 Second test case with ten devices: a generator, a storage device, three wind turbines and five consumers of energy

6 Summary

The simple auction algorithm applied to the Agent-based Power Balancing System performs well. Moreover, in the simple case considered it assures imbalance reduction. The main difficulty is to deal with the multiple, concurrent auction processes, initiated by different agents. However, the results presented show that the system performs well, with short imbalance reduction time intervals. As the system runs multiple auction processes, it is tempting to apply the multi-commodity market model (M^3) [6] to organize the notation. However, the impact of using the XML notation on prolongation of the communication time have to be checked. Further development and examination of the Agent-based Power Balancing System is therefore necessary.

Acknowledgements. The research was supported by the Polish Ministry of Science and Higher Education under the grant N N519 580238, and by the Foundation for Polish Science under International PhD Projects in Intelligent Computing. Project financed from The European Union within the Innovative Economy Operational Programme 2007-2013 and European Regional Development Fund.

References

1. Abbey, C., Joos, G.: Energy management strategies for optimization of energy storage in wind power hybrid system. In: IEEE 36th Power Electronics Specialists Conference, PESC 2005, pp. 2066–2072. IEEE (2005)
2. Allweyer, T.: BPMN 2.0: Introduction to the Standard for Business Process Modeling. Herstellung und Verlag: Books on Demand GmbH (2009)

3. Bellifemine, F.L., Caire, G., Greenwood, D.: *Developing Multi-Agent Systems with JADE*. John Wiley Sons Ltd. (2007)
4. Bertsekas, D.P.: Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications* 1(1), 7–66 (1992)
5. Dimeas, A.L., Hatziargyriou, N.D.: Operation of a multiagent system for microgrid control. *IEEE Transactions on Power Systems* 20(3), 1447–1455 (2005)
6. Kaleta, M., Traczyk, T. (eds.): *Modeling Multi-commodity Trade: Information Exchange Methods*. AISC, vol. 121. Springer, Heidelberg (2012)
7. Lagorse, J., Simões, M.G., Miraoui, A.: A multiagent fuzzy-logic-based energy management of hybrid systems. *IEEE Transactions on Industry Applications* 45(6), 2123–2129 (2009)
8. McArthur, S.D.J., Davidson, E.M., Catterson, V.M., Dimeas, A.L., Hatziargyriou, N.D., Ponci, F., Funabashi, T.: Multi-agent systems for power engineering applications part i: concepts, approaches, and technical challenges. *IEEE Transactions on Power Systems* 22(4), 1743–1752 (2007)
9. McArthur, S.D.J., Davidson, E.M., Catterson, V.M., Dimeas, A.L., Hatziargyriou, N.D., Ponci, F., Funabashi, T.: Multi-agent systems for power engineering applications part ii: technologies, standards, and tools for building multi-agent systems. *IEEE Transactions on Power Systems* 22(4), 1753–1759 (2007)
10. Nahorski, Z., Pałka, P., Radziszewska, W., Stańczak, J.: Założenia dla systemu wieloagentowego do bieżącego bilansowania energii generowanej i pobieranej. Tech. rep., RB/61/2011, Systems Research Institute, Polish Academy of Science (2011)
11. Nahorski, Z., Radziszewska, W.: Ogólny projekt systemów bilansowania energii w ośrodku badawczo-szkoleniowym. Tech. rep., RB/77/2011, Systems Research Institute, Polish Academy of Science (2011)
12. Nahorski, Z., Radziszewska, W., Parol, M., Pałka, P.: Intelligent power balancing systems in electric microgrids. *Rynek Energii* 1(98), 59–66 (2011)
13. Palma-Behnke, R., Benavides, C., Aranda, E., Llanos, J., Saez, D.: Energy management system for a renewable based microgrid with a demand side management mechanism. In: 2011 IEEE Symposium on Computational Intelligence Applications In Smart Grid, CIASG, pp. 1–8. IEEE (2011)
14. Ricalde, L.J., Ordonez, E., Gamez, M., Sanchez, E.N.: Design of a smart grid management system with renewable energy generation. In: 2011 IEEE Symposium on Computational Intelligence Applications In Smart Grid, CIASG, pp. 1–4. IEEE (2011)
15. Schoeneich, R.O., Pałka, P.: Application of auction mechanisms in routing decisions for delay and disruptive tolerant networks. *Przegląd Telekomunikacyjny- Wiadomosci Telekomunikacyjne LXXXIV(8-9)*, 1000–1003 (2011)
16. Shoham, Y.: Agent oriented programming. *Artificial Intelligence* 60(1), 51–92 (1993)
17. Shoham, Y., Leyton-Brown, K.: *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press (2009)
18. Smith, R.G.: The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers* 100(12), 1104–1113 (1980)
19. Tsikalakis, A., Hatziargyriou, N.: Centralized control for optimizing microgrids operation. In: 2011 IEEE Power and Energy Society General Meeting, pp. 1–8. IEEE (2011)
20. Vogt, H., Weiss, H., Spiess, P., Karduck, A.P.: Market-based prosumer participation in the smart grid. In: 2010 4th IEEE International Conference on Digital Ecosystems and Technologies, DEST, pp. 592–597. IEEE (2010)
21. Wooldridge, M., Bussmann, S., Klosterberg, M.: Production sequencing as negotiation. In: *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM 1996*, pp. 709–726 (1996)

Multi-commodity Trade Application to the Routing Algorithm for the Delay and Disruptive Tolerant Networks

Piotr Pałka and Radosław Schoeneich

Abstract. The paper considers analysis of the specific routing problem, concerning the delay and disruptive tolerant networks (DTN). Those networks are characterized by their lack of connectivity – there may be no connection between any pair of nodes. Moreover, the nodes are mobile, and the network configuration changes over time. The simulation tool that is used to analyzing particular routing methods is presented. The simulation tool is based on the Multi-commodity Market Model (abbrev. M^3), and implemented according to the multi-agent programming paradigm [19]. Thus, the paper also describes one of the M^3 applications. Moreover, the preliminary results of applied Random Walk, and Auction-based routing algorithms for the DTN are also presented.

1 Introduction

Delay and Disruptive Tolerant Networks (DTN), first introduced by K. Fall [10], are emerging as a promising extensions to improve communications in mobile, wireless, ad-hoc networks. The wireless ad-hoc DTN is composed of nodes which are intermittently connected in a spontaneous manner. The changeable structure of the network and high mobility of nodes are reasons of network partitions. The property of permanent network partitions causes that it is impossible to establish the data packet source-destination path. Therefore the communication in the DTN is done by ferrying data packets using mobility of nodes. This idea is based on an assumption that each node receives, stores, and exchanges packets in every opportunity with other

Piotr Pałka

Warsaw University of Technology, Institute of Control and Computation Engineering,
Warsaw, Poland

e-mail: P.Palka@ia.pw.edu.pl

Radosław Schoeneich

Warsaw University of Technology, Institute of Telecommunications, Warsaw, Poland

e-mail: rschoeneich@tele.pw.edu.pl

nodes. This style of working is called a store-carry-forward paradigm. The nodes which are storing and transporting the packets are called message ferries or mules.

Although a store-carry-forward mechanism is well known in the human environment (e.g. post and courier services), it cannot be easily implemented in an environment of small and mobile wireless networks. Main important problems are resource limitations of the communication, and nodes storing capability. Therefore the proper routing algorithm, which saves restricted node resources is fundamental in the DTN network creation.

Due to tolerant for intense network disruptions the DTN has many potential applications. A DTN can be applied in rescue and emergency situations (SAR), in cases where traditional wired and wireless communication infrastructure is unserviceable e.g. in earthquake scenarios, in small-populated and hardly-accessible terrains, in railway and metro tunnels, or in the warfare missions.

Proposed in the paper packet routing algorithm for the DTN, assumes application of the mechanism theory to determine choice of the node, which will be the best ferry for appropriate data packets. It is assumed that the memory limit of particular node is usually lesser than the summary size of all packets that exists in the network. In such a situation, the replication principle [2] is useless, and the more memory saving forwarding principle is applied. In such a situation, nodes should take a decision whether to accept the packet and become the ferry or not. Also, the node which passes the packet has to decide of to which nodes it should forward the packet, in order to obtain fastest packet delivery. Thus, the proposition of the Auction-Based Routing Algorithm for DTN networks (abbrev. ABRA) is done [18]. ABRA assumes that the auction process is performed due to take a decision about to which node the particular packet should be forwarded. The operator of such an auction is passing packet node, the auctioneers are the nodes in the vicinity of operator, which agree to become a mule.

The contributions of this paper are fourfold: (i) proposition of auction-based routing algorithm for the delay and disruptive tolerant network, (ii) development of the communication protocol for the auction-based routing algorithm (iii), implementation of the simulation tool, using the multi-agent system and M³ model, and (iv) evaluation of proposed auction-based routing algorithm on the preliminary tests.

2 Related Work

Presented idea of the routing algorithm for DTN is related to two problems: traditional DTN routing algorithms and social bahaviour of wireless nodes. In context of our idea the important backgrounds are: the simplest and most universal Epidemic Routing (abbrev. ER) [24] where routing is done by flooding messages in whole network, and Random Walk (abbrev. RW) [9], which is more selective in number of nodes engaged in routing messages. The ER is the most effective in terms of the delivery ratio, but it is also very expensive. One of the way of reducing costs of message delivery is to reduce the number of replicated messages, and nodes engaged to

route messages. The simplest way to achieve this is propagate only one copy of the message (so called forwarding principle [2]) without any routing strategies, e.g. randomly like in RW, but this way occurs low delivery ratio. The solutions which are compromise in number of copies (so called replication principle [2]) and number nodes engaged are much more effective. The solutions like Sprite and Wait and Sprite and Focus [21, 23, 22] are now almost classical reference.

Nodes mobility is useful in DTN routing strategies, depending on the type of the node and its role in the network, the node can contact not only with different numbers of nodes, but also with their various types. Based on this relationship, it can be selected the points of concentration (so-called hub nodes), where a higher density of nodes, and thus a greater potential encounters and exchanges messages between them. This property was used in Island Hopping [17] MobySpace [14], MV [5], as well as the Prophet [15]. This observation was the source of many works based on the study of social ownership of the nodes in DTN networks. Example is the ECCO-PS [25] – the overlay solution for publish-subscribe, adapted to transmit messages described content. The other solutions that use the social ownership of the network is HiBOP [4], SimBet [8], OnMove [7], and SocialCast [6], where the authors proposed a solution to learning, based on the relationship of nodes in social networks. Although in all this works there is phase of exchange of information, there is no advanced negotiation known in context of theory of auctions, therefore at the best of authors knowledge the presented idea of routing strategy in DTN networks is novel.

3 Auction-Based Routing Algorithm

The ABRA algorithm also assumes learning of the nodes. The decision of, to whom the data packed should be forwarded, is taken according to an auction result. The ABRA algorithm from the market mechanism viewpoint, can be seen as the multiple, one-side, multi-commodity auctions on the particular packets, performed in the distributed fashion. Every node can be an auction operator, and also every node can act as the offerer. Moreover, no constraints on the exchanged commodities occurs in the trade, besides the obvious one, that is the only those nodes participate in auction that are close enough to be able to communicate with the operator node. In opinion of the authors, applying the methodology based on market mechanisms design to the routing algorithm creation, will result in better packets routing, especially taking into account limited resources of particular nodes (e.g. memory), and limited time in which the packets should be delivered. The proposition of the auction-based routing protocol for DTN appears first in the [18].

Particular group of the nodes can perform a packet forwarding, if the group is in the communication distance. In the further parts of the paper such a situation will be called a node meeting. Now, let us introduce some notation. Let n be a node chosen from the set N of all nodes in the DTN. The $m \in 1, \dots, M$ is the number of particular node meeting. The meeting occurs, when the number of nodes gathers in the

communication distance, and are allowed to the packets exchange. Let the $G \subseteq N$ be a group of nodes that lie at a allowing communication distance. Let $|G|$ indicate the number of nodes in the group. Now, let $G_m : |G_m| > 1$ be a group that meets while the m -th meeting. The c is the single packet from the set of all packets C . The $k \in G_m \subseteq N$ be a node index that participate in the m -th meeting. Now, let us denote by $C_{mk} \subseteq C$ the set of packets owned by the k node during the m -th meeting. Note, that under the assumption of packets forwarding, the following condition occurs $\forall_{m \in 1, \dots, M} \forall_{k, l \in G_m, k \neq l} C_{mk} \cup C_{ml} = \emptyset$, thus the sets of packets of the meeting nodes are always disjoint. However, under the assumption of packets replication, the following condition occurs: $\forall_{m \in 1, \dots, M} \forall_{k, l \in G_m, k \neq l} C_{mk} \cup C_{ml} \subseteq C$. This means that particular packet can be replicated, and multiple copies of it can meet. After each meeting m , the appropriate routing protocol takes place. It is composed of four steps: (i) broadcast, (ii) offering, (iii) direct delivery and allocation, and (iv) forwarding. Note that those steps are not synchronized among the nodes. The detailed description of those steps is described as follows.

Broadcast. When a node $k \in G_m$ discovers change in its vicinity (e.g. approach of a group of nodes), it broadcasts an informations about the packets C_{mk} that are intended to pass by it. Simultaneously, each node $k \in G_m$ waits for incoming broadcasts, and collects the packets information. Note, that each node contemporary is an auction operator, and offerer of the other auctions. The communicate comprising the list of packet information L_{mk} is called call for proposal (CFP), as set forth in papers regarding the communication during the auctions [20, 11]. Each element of the list $l \in L_{mk}$ comprises: identifier of the receiver, TTL (time to live), and size of the packet.

Offering. During this step, nodes respond to the CFP communicate, by sending appropriate offers to the suitable auction operator. Each node $j \in G_m$ responds to the the CFP of the k -th node by sending the communicate O_{mjk} . Note that for each O_{mjk} the $k \neq j$ inequality occurs. It comes from the assumption that the node does not answer for its CFP. The offers $o \in O_{mjk}$ comprise the packets that adequate node is willing to obtain. The main parameter of the offer is the purchase price p_o . The price is modeled as the appropriate metric, and the assumption is made that the greater the metric is, the more suitable the node is for the task of being the ferry of the corresponding data packet. Thus, the suitable metric should be assumed. We assume the metric proposed in [18].

Direct delivery. During this step, each node that received CFP communicate, tries to match the appropriate packet receivers with the identifier of the node that sends CFP (thus the nodes in its vicinity). If succeeds, it matches appropriate packets as intended to deliver. For each operator $k \in G_m$ the set $I_{mk} \subseteq G_mk$ is created, which contains the packets intended to direct delivery. The packets from this set are not further allocated.

Allocation. After each operator $k \in G_m$ gathers send to him offers O_{mjk} , the appropriate allocation is performed, using appropriate market mechanism allocation rule. In simplest case it is an simple multi-commodity, sealed-bid, one-side auction, namely the winner of the packet (thus the node that become the ferry of the

particular data packet) $c \in B_{mk} \setminus I_{mk}$ is the node that submitted offer with highest price $\hat{p}_o = \arg \max_{o \in O_{mjk}} p_o$. It is possible that the operator introduces the minimal price, and only offers with higher prices are considered. The multi-commodity of the auction, result from the fact, that every data packet destined to particular node is unique, in the sense of different metric assigned. Thus the number of commodities in the auction is equal to the number of the packet receivers.

Forwarding. At the end of the ABRA protocol, the forwarding of appropriate directly delivered and forwarded packets takes place. For the sake of energy efficiency, it is assumed that each of the nodes, sends bundle of the packets to each of destined recipients and ferries.

4 Application of M^3 Model to the Routing Algorithm

The Multi-commodity Market data Model (abbrev. M^3) is a set of formal data models, which results in the XML-derived information interchange specification. It is a proper data model for many complex markets [13, 12]. The multi-commodity trade concerns widely recognized infrastructure markets, with complex infrastructure and security constraints, e.g. electricity energy markets, bandwidth trading in telecommunication networks, allocation of railway resources. It not only makes use of bundles of commodities, but also allows for complex bidding processes. Also, both centralized trading on the exchange platform, as well as bi- or multilateral negotiating contracts on the distributed market are supported. Since ABRA bases on the mechanism theory, it is tempting to apply the M^3 model notation at least for the part responsible for the auction procurement. Thus, let us consider particular elements of the M^3 notation.

Call for proposal. This communicative act is used to report willingness to trade with packets by the node which noticed a change in its vicinity. This communicate should contain: a list of unique identifier of packets to forward, the size of particular packets, their time to live parameter, and the receiver name. To formulate such a communicate, we need to decide on its data structure. From the viewpoint of the M^3 model application, the packet is treated as the trading commodity. The particular parameters of the packet can be treated as the generic parameters. Thus, we propose the following notation of single packet information, using the M3XML notation [13]. Before the packet information structure will be presented, it is essential to introduce the commodity kind structure. The commodity kind represent the metadata for the commodity (see Listing 1). Having commodity kind defined, it is possible to formulate the exemplary structure of the packet information (see Listing 2).

Submitting offers. The offer should contain the unique identifier of corresponding commodity (e.g. data packet), and the price that the purchaser is willing to pay for it. Note that we do not consider data packets trading, but the effective packet allocation, and the price helps us to determine how fast, and reliable particular nodes are able to deliver the packet to the receiver. In the Listing 3 the exemplary offer is notated.

Listing 1. Exemplary M3XML fragment describing metadata for the single packet information – commodity kind

```

<m3:commodityKinds>
  <m3:CommodityKind category="packet" id="dtn:packet">
    <m3:name>Message sent in the DTN network</m3:name>
    <m3:typeParameter dref="dtn:ttl"/>
    <m3:typeParameter dref="dtn:size"/>
    <m3:typeParameter dref="dtn:receiver"/>
  </m3:CommodityKind>
  <m3:ParameterDefinition id="dtn:ttl" dataType="xsd:long" unitOfMeasure="s">
    <m3:name>Time to live</m3:name>
  </m3:ParameterDefinition>
  <m3:ParameterDefinition id="dtn:size" dataType="xsd:long" unitOfMeasure="B">
    <m3:name>Size of the packet</m3:name>
  </m3:ParameterDefinition>
  <m3:ParameterDefinition id="dtn:receiver" dataType="xsd:QName">
    <m3:name>Message receiver</m3:name>
  </m3:ParameterDefinition>
</m3:commodityKinds>

```

Listing 2. Exemplary M3XML fragment describing the single packet information – commodity

```

<m3:commodities>
  <m3:Commodity id="dtn:4365129341" dref="dtn:packet">
    <m3:description>Message to the node9</m3:description>
    <m3:parameter dref="dtn:ttl">1294</m3:parameter>
    <m3:parameter dref="dtn:size">1024000</m3:parameter>
    <m3:parameter dref="dtn:receiver">node9</m3:parameter>
  </m3:Commodity>
</m3:commodities>

```

Listing 3. Exemplary M3XML fragment describing the offer

```

<m3:offers>
  <m3:Offer id="dtn:off_node7_5431289" offeredPrice="50.0">
    <m3:description>Offer submitted by the node7</m3:description>
    <m3:offeredBy ref="dtn:node7"/>
    <m3:volumeRange minValue="1" maxValue="1"/>
    <m3:ElementaryOffer>
      <m3:offeredCommodity shareFactor="-1" ref="dtn:4365129341"/>
    </m3:ElementaryOffer>
  </m3:Offer>
</m3:offers>

```

5 Multi-agent Simulation Tool

As the particular nodes take autonomous decisions about to whom forward a packet, those nodes can be treated as the autonomous entities. Moreover, the nodes communicate with one another in order to forward the packets in the best possible way. Note, that no communication can be made without message passing between particular nodes. Thus, for the sake of simulation of the DTN, the decision taking nodes can be treated as the agents, and the DTN, along with all nodes, can be modeled as

the multi-agent system [20]. Main agent role to be modeled is the DTN Node agent. It is the representation of the real mobile node from the disruptive and delay tolerant network. It is assumed that such a node can travel through some area, and gather the information, that is formed into the data packets destined to other DTN nodes. Also, what is probably most important from the viewpoint of the paper, DTN nodes route packets according to given routing algorithm (ABRA). The diagram of the communication protocol is presented using the BPMN 2.0 notation [11]. The notation was used, due to its convenience and transparency.

When the group of nodes G is constituted, the following process occurs (see Fig. 1). First, the meeting manager is waiting for a meeting. Every time when it comes to the meeting, two new parallel behaviors (behavior is the element of agent acting) are created: auction operator behavior and the offerer behavior. As can be seen in the Fig. 1, the DTN Node agent communicates with other DTN node agents, and tries to forward its packets in the best possible way. The auction operator, which is responsible exactly for selecting the best mules for the data packets, first prepares the packets information, and then broadcasts those information among the nodes. This is essentially the broadcast phase. Afterwards, the auction operator behavior collects all the offers that was submitted in the response of given broadcast. For the sake of possible communication loss, the timeout is introduced. When either all offers are collected or the timeout occurs, the DTN Node assigns the packets that can be passed directly to their receivers (direct delivery phase), and allocates the rest of the packets. As have been told before, the allocation is performed according to appropriate allocation rule. Finally, the packets are distributed according to allocation and direct association results. The offerer behavior is waiting for the packet information, formulates the offers, sends them back to the suitable DTN node, and finally is waiting for the allocation results. Note, that the direct delivery phase takes place after the broadcast phase. This is due to the fact, that particular nodes do not know

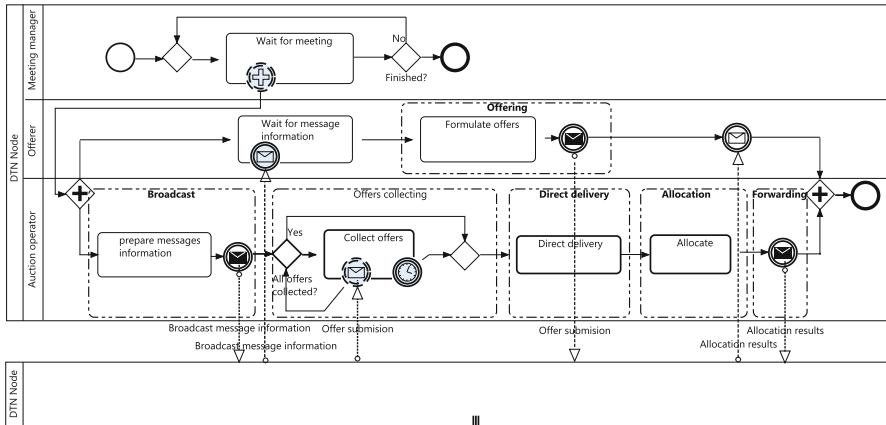


Fig. 1 BPMN collaboration diagram presenting communication among DTN nodes during the auction-based routing algorithm

the identifiers of the nodes in their vicinity. Thus, the broadcast phase is equivalent to the hello communicate, and each node that notice change in its proximity should send a broadcast packet (even if it have not any packets to forward).

6 Preliminary Results

The simulation environment is implemented using JADE 4.0 framework [3], and the Java 1.6 language. Moreover, the simulator uses the M³ notation, and to facilitate its usage, the JAXB framework [16] that enables Java classes building from XML schemes (i.e. M3XML documents) is used.

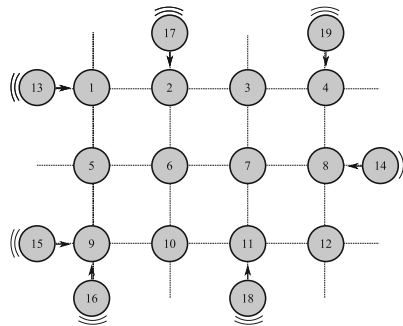


Fig. 2 Preliminary data: urban network with 3x4 stationary, and 3+4 mobile nodes. Dotted lines represent routes of the mobile nodes.

The simple test case concerns a urban network, with 12 nodes placed in the vertices of the 3x4 grid, and 7 mobile nodes, traveling across the grid (see Fig. 2). The case study concerns that in each node 10 packets are generated (during the simulation). Recipient of each of the packet is also generated randomly. Thus, 190 packets are generated. Time to live parameter is also drawn from the uniform distribution $U(150\ 000, 200\ 000)$. For the sake of simulation, it is assumed that TTL is counted in milliseconds. The comparative tests was performed for one of simplest routing algorithm – Random Walk [21]. The tests was performed in order to validate ABRA, especially to check the correctness of the communication protocol of the auction. For ABRA protocol 330 ± 21 , and for Random Walk 306 ± 19 meetings was held, and for each meeting 2-3 auction processes (the number results from the grid structure) was held. Each auction process was performed properly.

To compare the quality of ABRA and Random Walk routing algorithms, the following characteristics are analyzed: average time to deliver packets to recipients (AT); packet delivery ratio (DR); and average number of the storage nodes per packet (SNN). The results averaged for one hundred executions are presented in table 1.

Table 1 Preliminary data for one hundred simulations of DTN, for two routing protocols

	ABRA		Random Walk	
	avg.	std. dev.	avg.	std. dev.
AT[ms]	37 954.4	39 925.1	31 102.2	38 443.0
DR[%]	70.79	6.00	42.14	5.33
SNN[pcs.]	8.44	8.28	8.80	8.92

As can be seen in table 1, ABRA algorithm has a better delivery ratio, and the average number of participating ferry nodes in carrying single message is a bit smaller. The average time of delivery is greater in the case of ABRA, however, it must be stressed, that more data packets was delivered in the case. Note, that the not delivered packets was removed from particular nodes (to free their memory).

7 Conclusion

The paper presents proposition of auction-based routing algorithm for the delay and disruptive tolerant network. Also the development of the communication protocol for the auction-based routing algorithm is presented. The correctness of both auction-based routing algorithm, and the communication protocol for this algorithm, has been validated, using the implementation of the simulation tool. The validation seems to be correct, i.e. ABRA routing algorithm works according to the store-carry-forward paradigm, moreover, it gives better results (e.g. delivery ratio), than the Random Walk algorithm. The communication protocol, also is correctly developed. Thus, the ABRA appears to be a good routing algorithm for delay and disruptive tolerant network, and it will be further developed.

Acknowledgements. The research was supported by the Polish Ministry of Science and Higher Education under the grant N N514 044438.

References

1. Allweyer, T.: BPMN 2.0: Introduction to the Standard for Business Process Modeling. Herstellung und Verlag: Books on Demand GmbH (2009)
2. Balasubramanian, A., Levine, B.N., Venkataramani, A.: Replication routing in dtns: A resource allocation approach. *IEEE/ACM Transactions on Networking* 18(2), 596–609 (2010)
3. Bellifemine, F.L., Caire, G., Greenwood, D.: *Developing Multi-Agent Systems with JADE*. John Wiley Sons Ltd. (2007)
4. Boldrini, C., Conti, M., Jacopini, J., Passarella, A.: Hibop: a history based routing protocol for opportunistic networks. In: *World of Wireless, Mobile and Multimedia Networks*, pp. 1–12 (2007)

5. Burns, B., Brock, O., Levine, B.N.: Mv routing and capacity building in disruption tolerant networks. In: Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2005, vol. 1, pp. 398–408 (2005)
6. Costa, P., Mascolo, C., Musolesi, M., Picco, G.: Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications* 26(5), 748–760 (2008)
7. Cuevas, R., Jaho, E., Guerrero, C., Stavrakakis, I.: Onmove: a protocol for content distribution in wireless delay tolerant networks based on social information. In: Proceedings of the 2008 ACM CoNEXT Conference, CoNEXT 2008, pp. 40:1–40:2 (2008)
8. Daly, E.M., Haahr, M.: Social network analysis for routing in disconnected delay-tolerant manets. In: Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2007, pp. 32–40 (2007)
9. Dhillon, S.S., Mieghem, P.V.: Comparison of random walk strategies for ad hoc networks. In: Proceedings of Sixth Annual Mediterranean Ad Hoc Networking Workshop, pp. 196–203 (2007)
10. Fall, K.: Messaging in difficult environments. Tech. rep., IRB-TR-04-019 Intel Corp. (2004)
11. Foundation for Intelligent Physical Agents, <http://fipa.org/>
12. 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)
13. Kaleta, M., Traczyk, T. (eds.): Modeling Multi-commodity Trade: Information Exchange Methods. AISC, vol. 121. Springer, Heidelberg (2012)
14. Leguay, J., Friedman, T., Conan, V.: Evaluating mobility pattern space routing for dns. In: Proceedings of IEEE INFOCOM 2006 (2006)
15. Lindgren, A., Doria, A., Schelén, O.: Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communications Review* 7(3), 19–20 (2003)
16. McLaughlin, B.: Java and XML data binding. O'Reilly & Associates Inc. (2002)
17. Sarafijanovic-Djukic, N., Pidrkowski, M., Grossglauser, M.: Island hopping: Efficient mobility-assisted forwarding in partitioned networks. In: 2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON 2006, pp. 226–235 (2006)
18. Schoeneich, R.O., Pałka, P.: Application of auction mechanisms in routing decisions for delay and disruptive tolerant networks. *Przegląd Telekomunikacyjny- Wiadomosci Telekomunikacyjne LXXXIV(8-9)*, 1000–1003 (2011)
19. Shoham, Y.: Agent oriented programming. *Artificial Intelligence* 60(1), 51–92 (1993)
20. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press (2009)
21. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Single-copy routing in intermittently connected mobile networks. In: IEEE Sensor and Ad Hoc Communications and Networks, pp. 235–244 (2004)
22. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In: Pervasive Computing and Communications Workshops, pp. 79–85 (2007)
23. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, WDTN 2005, pp. 252–259 (2005)
24. Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Tech. rep., CS-2000-06, CS Dept., Duke University (2000)
25. Yoneki, E., Hui, P., Chan, S., Crowcroft, J.: A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In: Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, MSWiM 2007, pp. 225–234 (2007)

Offers Discovery and Identifying User Requirements for Multi-commodity Trade in Open Markets

Dominik Ryzko and Anna Wróblewska

Abstract. The paper describes a novel approach to discovery of offers in multi-commodity trade on open markets. Methods for gathering user requirements and for offer search are described. The process utilizes semantic technologies and multi-agent architecture. Examples are shown in the domain of trading electric energy with the use of M^3 ontology.

1 Introduction

Discovery of interesting offers is a crucial task in e-commerce. In case of a close well defined environments, the task can be reduced to search through a catalog of similarly structured entries. However, in open environments like the Internet, the task is far from simple. Various catalogs provide specific forms of description. Similarly the semantics of offers can differ from host to host. These problems are particularly visible in the case of complex products like the ones on multi-commodity markets. The product search problem is closely related to offer matching and both topics are very often researched together [5, 2, 14]. In this work we will concentrate on the process of interpretation of user requirements and search of relevant offers.

2 Research Problems and Existing solutions

Several papers have been published on discovery and matchmaking of offers for e-commerce. The most important approaches concentrate on the use of ontologies and

Dominik Ryzko · Anna Wróblewska
Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19,
00-665 Warsaw Poland
e-mail: [D.Ryzko, A.Wroblewska}@ii.pw.edu.pl](mailto:{D.Ryzko,A.Wroblewska}@ii.pw.edu.pl)

Description Logic or Web Services standards. In this paper we will concentrate on the former. More and more Multi-Agent paradigms are also influencing this field.

2.1 Product Search

The most simple approach to product search starts with keywords. This requires semantic interpretation of the user query, which is further used as a guideline for Semantic Search. Work in this area can be found in [11, 17]. Tran et al. base their approach on identifying two models. The first one is the Mental Model O_U , which corresponds to the information need that a user has in mind. The latter is the System Resource Model O_S , which corresponds to the system knowledgebase in the form of an ontology, used for answering queries. In such a setup the search process breaks down to translating the user query Q_U into the system query Q_S , which are expressed with the use of O_U and O_S respectively. Three stages are identified here:

- The elements in the user question Q_U are mapped to ontology elements from O_S
- further ontology elements are explored to better cover the initial information need in the mental model O_U
- the query Q_S is derived from the ontological representation achieved in the previous step

In [8] Preference XPATH is presented, a new XML-based search technology for e-commerce, that enables users to formulate complex customer or vendor preferences. Authors identify three main approaches for attribute search in XML:

- The naive approach of translating each condition preference into a hard selection.
- Translation of preferences into soft selection conditions and application of some ranked query model, using numerical scores and some weighted combination functions.
- Translation of preferences into soft selection conditions, modeling preferences qualitatively, e.g. as partial orders.

Such approach has several limitation due to the lack of semantics. More recently ontologies and Description Logic has been applied to address this issue. Offers can be unstructured or use ambiguous terminology. In the best case we can expect all offers to be using some kind of ontology. However these ontologies can differ significantly. Some parties can use M^3 while others GoodRelations [7] or some other more or less common solution. To tackle this obstacle ontology mapping technologies can be applied. This process is defined as 'The process that transforms the source ontology entities into the target ontology entities based on semantic relation. The source and target are semantically related at a conceptual level' [4].

SMART [2] is an example of a ontology based platform for searching and comparing products in e-commerce. However, instead of typical Description Logic, a fuzzy approach is proposed. Another approach is taken in [14], where authors devise Concept Abduction and Concept Contraction as non-monotonic inferences in Description Logics suitable for modeling offer matchmaking in a logical framework.

Colucci et al. [5] also propose the use of nonmonotonic inferences in a semantic matchmaking process. Here, the nonmonotonic approach allows to hypothesize missing attributes in the offers. Of course, this means some offers can match only partially, so offers are prioritized in order to propose direct matches first.

2.2 Using Intelligent Agents for Offer Discovery

Multi-agent systems (MAS) play an increasing role in the Electronic Commerce [6]. In a complex e-commerce setting a direct search through a catalog of offers is not possible. The process requires an intermediate step called Product Brokering. A broker is a party, which takes user needs, scans known product sources for matches and returns most promising results ordered by some relevance measure. MAS are especially well suited for handling this task in a distributed heterogeneous environment [9]. An example of agent platform for e-commerce can be found in [18], where GRAPPA framework is introduced. The system consists of a matchmaking engine for matching offers, a generic matchmaking library and a matchmaking toolkit.

An important area of research is devoted to Recommendation Agents (RAs), which recommend products based on implicit or explicit user preferences [21]. RAs use prediction algorithms to match items with user interests. They incorporate user-based or item-based algorithms or combination of both [15]. While RAs concentrate mainly on B2C market, several issues analyzed here can be useful also for more complex B2B trading. For example the problem of trust has been studied in [19].

3 Offer Discovery in Open Multi-commodity Markets

Multi-commodity markets impose significantly more advanced constraints on the product search than in the case of trading simple products. Complex structure of the objects results in much more difficult process of disambiguation. The discovery process can be broken into three distinct yet interconnected parts: formulating requirements, identifying offers and matching. In this section means for applying semantic technologies to facilitate offer discovery are proposed. An architecture for enabling efficient offer discovery is also introduced.

3.1 Semantic Technologies for Efficient Offer Discovery

3.1.1 Gathering User Requirements for the Discovery Process

As mentioned above, the first step of offer discovery is formulating requirements. The important aspect of this process is the language in which the user expresses the goals. We propose a two-stage approach. Firstly a search is performed through keywords. Then after preliminary keywords interpretation and matching them with

domain concepts, the process of specifying and clarifying of a query is started. Various users have different level of experience and different needs, therefore the second step can be omitted. However, there is a risk of discovering less relevant offers.

The first stage with keyword search requires analyzing the query and structuring a concept formulated in natural language into a formal description. Based on the approach by Tran et al. [17] presented above, we show how this can be done by translating keyword queries to Description Logic queries using background knowledge available in the M^3 ontology.

We have to map the keywords provided by users onto the concepts of the M^3 ontology. To this end, we need a lexical layer that will assign terms (words or phrases) to the conceptual layer (ontology). In the simplest way it can be accomplished with the use of a list of synonymous terms assigned to each ontology concept or property. Such onto-gazetteers are implemented in GATE NLP framework [1]. In more sophisticated methods we can apply a lexical layer like LEXO [20] that determine the adequate sense of keywords with the terms used in their contexts. Other linguistic layer, worth to mention, is LEMON [13] which uses syntactic behavior of terms to assess their meaning.

The second stage of formulating user requirements is a deeper interpretation of the query in order to build the adequate, full and disambiguated system query. In most cases the system needs to interact with the user to show additional relevant options for the query, and then to set all necessary data to detail the query. In the first case we can expand networks and commodities available on network nodes and arcs. We can show other kinds of commodities (subconcepts of a given commodity or other commodities at the same level of taxonomic hierarchy). In the second case of interaction with a user we can use restrictions and definitions of an Offer (see Fig. 2). We can expand all necessary properties and concepts being in relations with the base query concept (here Offer) in such a way that the new set composes a valid more detailed product description. Finally the system structuralized query will be composed, which will indicate search for complete offers matching user requirements, rather than only the elements missing from the original user description.

The following example illustrates the process described above. Let us assume a user requests a search for offers of Electric Energy in Warsaw. At first Electric Energy will be identified as a Commodity (a product) and Warsaw - as a Network Node (real or virtual). Then a user can be equipped with additional options to choose, e.g. (i) a map with all available network nodes and area (virtual) network nodes situated in Warsaw and around, (ii) a list of other commodities associated with electric energy (power reserves, certificates, derivative instruments, property rights). Then, in step two, additional ontology elements, which are necessary for a complete offer description will be identified. The additional elements are derived from object and data properties, which domain is an Offer, and from the concept definition. For example: an offer have to be in relation Market Entity which offers the product, Calendar Period in which the offer is valid etc. This ontology subset will be translated into query representing a particular offer type, with some of its elements instantiated representing constraints for the search process.

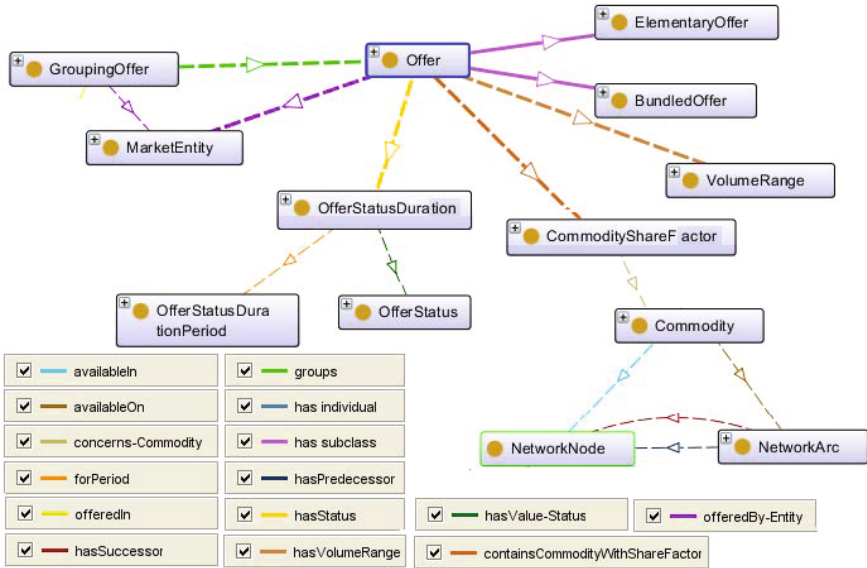


Fig. 1 Offers and related concepts, at the right: a legend with names of object properties

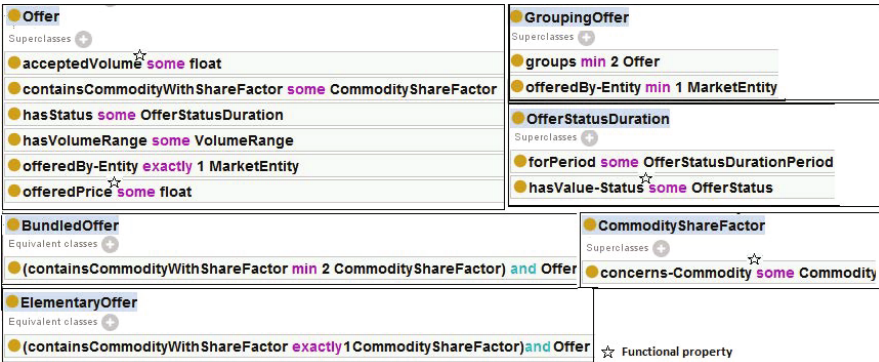


Fig. 2 Restrictions assigned to concepts used to express offers

Another possibility for formulating queries is to equip the user with an interface for inserting structured queries directly into the system [12, 16]. Such an interface can be generated from the M³ ontology. In this approach validation of queries against the knowledge base is ensured. This is particularly important in the case of multi-commodity trade, where definitions of products are often very complex. In detail this approach can be employed as follows. In the beginning the user makes general selection about the search. This should include selection of domain, market, simple vs bundled products etc. Based on this information the interface for query entry is generated in run time. Concepts from the ontology are, which compose an

offer are selected and their attributes are displayed as UI elements. If we take the case presented in the previous example, the interface will include fields for entering the Product, Network Node, Time, Price range, Market Entity offering the product etc. Wherever possible the list of values will be provided generated out of the concept instances stored in the ontology.

The result of both approach presented above is a structured query. Once we have the query structured as a DL query, the search process brakes down to the DL reasoning. One of several existing reasoners can be applied here. However, a nonmonotonic approach described in the previous section should be considered. While this increases computational complexity, it allows to reason with incomplete information, which would be impossible in a standard setting. Since the matching process itself is not the subject of this paper, we will not elaborate it here.

3.1.2 Offer Discovery from the Internet and Other Sources

On the other end of the search process, there is a need for discovery of offers, which can be matched against user queries. We will be concerned here only with the cases where the description is formed in a formalism other then the one used by the search system or it lacks any formal structure. The diagram [3](#) illustrates the discovery workbench for offer discovery from heterogeneous sources.

In the case of offer discovery from the Internet, selection of relevant web pages needs to be addressed first. When searching for specific information type, as considered here, focused crawling should be applied [\[3\]](#). To utilize this technique we will need a classifier, which will select for further crawling these pages which are likely to contain interesting offers. Such classifier has to be trained over a set of known examples of pages containing interesting offers. Obviously some negative examples have to be added to the training set as well. Since the focused crawling traverses the web along the link structure, it is important to seed the process with good starting points. This can be done by combining known sites containing relevant offers as well as search results from general purpose search engines generated by the use of relevant keywords describing the particular type of offers which are to be found.

After finding relevant pages containing offers the general web mining issues regarding parsing the content of the web pages remain to be solved. Separating the important content from the tags responsible for display and from irrelevant elements such as advertisements has to be conducted. To separate the extraction process from the interpretation and the use of the data we propose building a wrapper program, which is a common approach to data extraction from the web [\[10\]](#). With the wrapper program in place we can query any source of potential offers as a database.

The last part of the discovery process is the interpretation of the offers before they can be matched. If the steps described above are performed, we can use a general interpretation process abstract with respect to various sources of data. Similarly to the techniques described in the previous section regarding semantic search, the processing of offers requires the use of the M^3 ontology. Elements of the offer description should be mapped against a set of concepts from the ontology. Once these

concepts can be assembled into a complete offer, it can be extracted and saved. In several cases the matching process will fail this can be due to various reasons:

- Some of the offer elements were lost during the retrieval process
- An incomplete offer was published
- The description does not represent an offer

In the first of the cases described above improvement of the wrapper program is needed. In the second case it might be possible to receive a full offer by contacting the party responsible for the data. Depending on the system requirements such an offer can be discarded or presented to the user.

One specific case, that needs to be separately addressed, is the situation when a potential source of offers contains structured information, but its semantics is based on an ontology other than M^3 . In this case an ontology mapping process should be applied first, rather than trying to map each offer against the M^3 ontology. This initial effort, if successful, will pay off later when particular descriptions will be interpreted. To date several techniques for ontology mapping have been proposed [4]. We argue that the choice of this method is largely dependent on the specific features of the application domain and on the quality of the data sources. In any case this is mainly a manual or at best semi-automatic process, which results in creation of a mapping between the ontologies. Such a mapping does not have to be complete i.e. two ontologies can contradict each other, which results in a possibility for a partial interpretation of a particular offer. Once again specific requirements of a system should indicate the course of further actions.

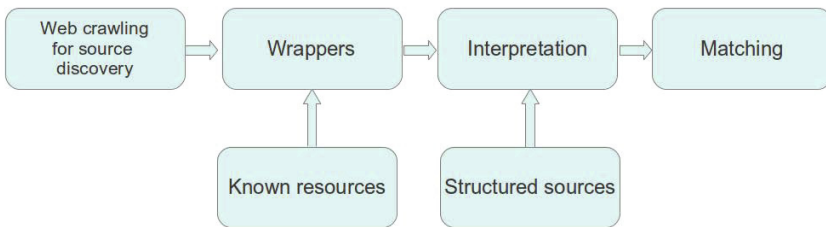


Fig. 3 Offer discovery

3.2 Architecture for Offer Discovery

The above description of the offer discovery process reveals a many-to-many relation between the users and the offers with high degree of heterogeneity of the system elements. Therefore a multi-agent approach to creating an architecture for offer discovery is proposed. We assume the search and matchmaking tasks will be carried out by the brokering agents. The agents will use explicit and implicit user preferences as a starting point for the search process. In the first case the user will formulate directly his requirements regarding desired products. In the latter case, a

personal agent will build a user profile based on user history. In either case the final outcome is a set of preferences, which have to be interpreted.

The Figure 4 illustrates the overall system architecture. In general three types of agents will be identified: Personal agents, Brokering agents and Harvesting agents.

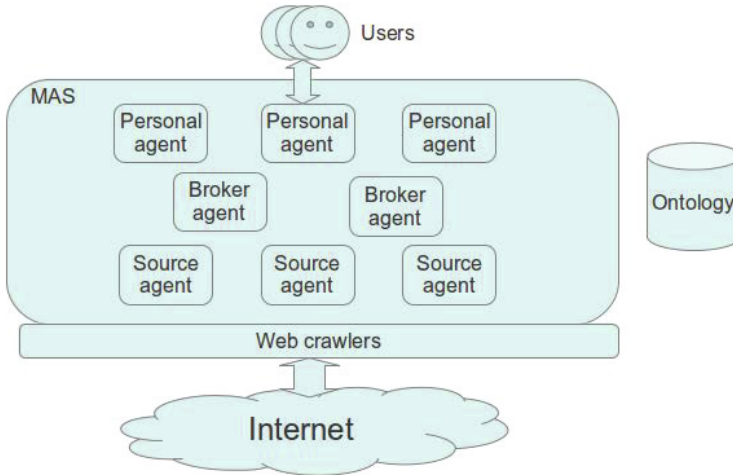


Fig. 4 Architecture

Personal agent will be responsible for interaction with the user and for building the user profile. When a query is received it is interpreted as described in Section 3. Firstly, it is processed to a query disambiguation module, which performs semantic interpretation by mapping query concepts onto the ontology concepts (TBox level). A mapped query is passed on to the disambiguation module, which in turn analyzes instances in the ontology (ABox level) in order to construct the final description. As described earlier in the paper this might involve interaction with the user in order to receive additional details. This is the reason for placing this functionality in the personal agent, which is the single entry point for user interface. In order to reduce user interaction a profile can be used which stores user preferences and can serve as a source of information for disambiguation. Once query interpretation is completed, it is passed on to the brokering agent. When results are retrieved, the personal agent presents them to the user. During this whole process a user profile is built and refined. Figure 5 shows the internal structure of a personal agent.

Brokering agent receives offer discovery request from various users. It selects known data sources relevant to a particular request and orders Harvesting agents to retrieve data from them. Brokering agent overlooks the processing of user requests and aggregates and sends back the results once they are ready. The aggregation process can be necessary in the case of search for complex products, composed of a set of more simple products, which cannot be provided by a single seller.

Harvesting agents are responsible for access to specific offer repositories. The architecture does not restrict any source types as long as the specific harvesting agent

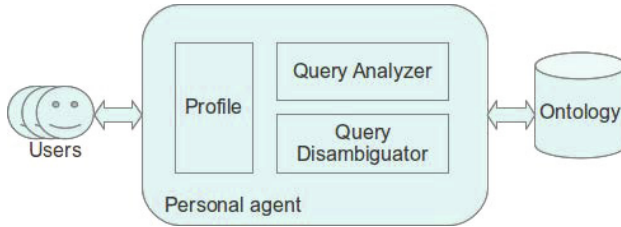


Fig. 5 Personal agent

knows the protocol to access the data. For example in case of a relational database the agent will use SQL to query and retrieve data. On the other end if the source is a subset of the Internet the agent will use a set of crawlers which access web pages via http protocol. Obviously knowledge of the protocol does not guarantee the proper interpretation of the data, so the techniques described in the previous section will be needed in order to manage the semantics of the retrieved offers.

4 Conclusions

This chapter deals with search for products in an open multi-commodity market. The task requires addressing several issues. This includes semantics of both offers and queries, which can be handled by the use of M^3 ontology and Description Logic. Another issue is distribution of trade in which heterogeneous parties can participate. This can be handled by distribution of computation and control among agents located in various network nodes. Future work might involve analysis of the approach against ontologies from other domains in order to verify its applicability.

References

1. General architecture of text engineering, <http://gate.ac.uk/>
2. Agarwal, S., Lamparter, S.: Smart: A semantic matchmaking portal for electronic markets. In: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology, CEC 2005, pp. 405–408. IEEE Computer Society, Washington, DC (2005)
3. Chakrabarti, S., van der Berg, M., Dom, B.: Focused crawling: A new approach to topic-specific web resource discovery. In: Proc. 8th Int'l World Wide Web Conf., pp. 1623–1640. Elsevier Science, New York (1999)
4. Choi, N., Song, I., Han, H.: A survey on ontology mapping. Newsletter ACM SIGMOD Record 35(3) (September 2006)
5. Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., Tinelli, E.: A non-monotonic approach to semantic matchmaking and request refinement in e-marketplaces. International Journal of Electronic Commerce 12(2) (2007)
6. He, M., Jennings, N.R., Leung, H.: On agent-mediated electronic commerce. IEEE Transactions on Knowledge and Data Engineering 15(4) (July/August 2003)

7. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 329–346. Springer, Heidelberg (2008)
8. Kießling, W., Hafenrichter, B., Fischer, S., Holland, S.: Preference XPATH: A query language for e-commerce. In: Proc. 5th Int. Konf. für Wirtschaftsinformatik, pp. 425–440 (2001)
9. Klusch, M., Sycara, K.P.: Brokering and matchmaking for coordination of agent societies: A survey. In: Coordination of Internet Agents: Models, Technologies, and Applications, pp. 197–224. Springer (2001)
10. Knoblock, C.A., Lerman, K., Minton, S., Muslea, I.: A machine-learning approach to accurately and reliably extracting data from the web. In: Proceedings of the IJCAI Workshop on Adaptive Text Extraction and Mining. (2001)
11. Lei, Y., Uren, V.S., Motta, E.: SemSearch: A Search Engine for the Semantic Web. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 238–245. Springer, Heidelberg (2006)
12. Liu, B., Chen, H., He, W.: Deriving user interface from ontologies: a model-based approach. In: Proc. ICTAI Tools with Artificial Intelligence (2005)
13. McCrae, J., Spohr, D., Cimiano, P.: Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 245–259. Springer, Heidelberg (2011), <http://data.semanticweb.org/conference/eswc/2011/paper/natural-language-processing/17>
14. Di Noia, T., Di Sciascio, E., Donini, F.M.: Semantic matchmaking as non-monotonic reasoning: A description logic approach. *Journal of AI Research* 29, 269–307 (2007)
15. Papagelis, M., Plexousakis, D.: Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. *Engineering Applications of AI* 18, 781–789 (2005)
16. Paulheim, H., Probst, F.: Ontology-enhanced user interfaces: A survey. *International Journal on Semantic Web and Information Systems* 6, 36–59 (2010)
17. Tran, T., Cimiano, P., Rudolph, S., Studer, R.: Ontology-Based Interpretation of Keywords for Semantic Search. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 523–536. Springer, Heidelberg (2007)
18. Veit, D., Muller, J., Schneider, M., Fiehn, B.: Matchmaking for autonomous agents in electronic marketplaces. In: Proc. Int. Conf. on Autonomous Agents 2001, pp. 65–66. ACM (2001)
19. Wang, W., Benbasat, I.: Trust in and adoption of online recommendation agents. *Journal of the Association for Information Systems* 6(3), 72–101 (2005)
20. Wróblewska, A., Protaziuk, G., Bembenik, R., Podsiadły-Marczykowska, T.: LEXO: a Lexical Layer for Ontologies - Design and Building Scenarios (2012)
21. Xiao, B., Benbasat, I.: E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Quarterly* 31(1), 137–209 (2007)

Fair Resource Allocation in Multi-commodity Networks

Tomasz Śliwiński

Abstract. The problem of fair resource allocation is of considerable importance in many applications. In this paper multiple commodity network flow combined with Ordered Weighted Averaging (OWA) aggregation operators and extensions is considered. The approach allows modeling various preferences with regard to flow distribution in a consistent and fairness-preserving way. It is shown that in this case OWA-based aggregation operators can be utilized just as easily as traditional lexicographic operators.

1 Introduction

A complex multi-commodity market mechanism is considered. Using network-based physical infrastructure for transportation, market participants want to exchange goods and services competing for limited resources. Such market mechanisms are known in the literature (see [5]), but only limited attention is paid to the problem of fairness of resource distribution among the participants.

There is no common definition of fairness. Intuitively, one wants the uniform criteria to be treated equally and impartially. For example, in the multiple users system resources can be allocated in a way that each user gets the same outcome. Obviously, this approach becomes inefficient when, for some reason, one of the users can get only very small value because then everyone gets the same small value.

One of the ways to overcome this difficulty is the application of the so called Max-Min Fairness (MMF) concept. This allows to achieve not only fair solutions but also decently efficient in terms of the system utilization. Fairness is accomplished by simple max-min optimization with regularization through maximization of the

Tomasz Śliwiński

Warsaw University of Technology Institute of Control & Computation Engineering,
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland

e-mail: tswiwin@elka.pw.edu.pl

second largest outcome (provided that the smallest one remains as large as possible), maximization of the third smallest (provided that the two smallest remain as large as possible), and so on. This approach prevents some demands with structurally low throughputs from blocking/disabling the max-min function. This is, however, a stiff approach that usually does not allow any other criteria, the overall efficiency (total throughput) in particular. Moreover, it requires sequential repeated optimization of the original problem.

Other commonly used approach is based on two criteria optimization with total throughput as one and some fairness measure as the other criterion. Applying weights one can to some extent express preferences with regard to either objective.

In this paper we investigate the application of the Ordered Weighted Averaging (OWA) aggregation with extensions as consistent, reasonable and fairness-preserving approach to modeling various preferences (from the extreme pessimistic, through neutral to extreme optimistic) with regard to outcomes distribution in multiple commodity network flow problems.

In the OWA aggregation ([15], [16]) the weights are assigned to the ordered values (i.e., to the largest value, the second largest and so on) rather than to the specific criteria. The OWA operator provides a parameterized family of aggregation operators, which include many of the well-known operators such as the maximum, the minimum, the k-order statistics (including CVaR), the median and the arithmetic mean. The OWA satisfies the properties of strict monotonicity, impartiality and, in the case of monotonic increasing weights, the property of equitability (satisfies the principle of transfers – equitable transfer of an arbitrary small amount from the larger outcome to a smaller outcome results in a more preferred achievement vector). Thus the OWA-based optimization generates the so-called equitably efficient solutions (cf. [6] for the formal axiomatic definition). According to [6] and [12], equitable efficiency expresses the concept of fairness, in which all system entities have to be treated equally and in the stochastic problems equitability corresponds to the risk aversion [3]. Since its introduction, the OWA aggregation has been successfully applied to many fields of decision making [17, 16, 7]. When applying the OWA aggregation to multicriteria optimization, the weighting of the ordered outcome values causes that the OWA optimization problem is nonlinear even for linear programming formulation of the original constraints and criteria. Yager [15] has shown that the nature of the nonlinearity introduced by the ordering operation allows one to convert the OWA optimization into a mixed integer programming problem. We have shown [9] that the OWA optimization with monotonic weights can be formed as a standard linear program of higher dimension. Its significant extension introduced by Torra [13] incorporates importance weighting into the OWA operator forming the Weighted OWA (WOWA) aggregation as a particular case of Choquet integral using a distorted probability as the measure. The WOWA averaging is defined by two weighting vectors: the preferential weights and the importance weights. It covers both the weighted means and the OWA averages as special cases. Some of the example applications of importance weights include definition of the size or importance of processes in a multi-agent environment, setting scenario probability (if uniform objectives represent various possible values of the same uncertain outcome

under several scenarios), or job priorities in scheduling problems. In [10] we have shown that in the case of monotonic preferential weights also WOWA aggregation can be modeled by linear extension of the original problem.

2 Multiple Commodity Flow Optimization

Let us consider a network G consisting of a set V of nodes and of a set E of undirected links, each with given capacity c_e ($e \in E$). There is also a set $J = \{1, 2, \dots, m\}$ of services defined in the network. Each service $j \in J$ depends on a flow between the given pair of nodes. The flow can be routed simultaneously on several paths chosen from a set M_j of paths allowed for each service (bifurcated flow). Services can utilize any bandwidth assigned. This can be found in Voice over IP or Video on Demand services where bigger bandwidth means better audio or video quality. The objective is to allocate link capacities (common resource) to competing services maximizing the total throughput and taking care of fairness of the allocation.

Let q be an index of a feasible flow path for service j , $p \in M_j$. The path is defined by input parameters δ_{ejp} ($e \in E, j \in J, p \in M_j$) (equals 1 if and only if link e belongs to path p of service j). Denote the flow assigned to path p by f_{jp} and the total flow for service j by f_j . The basic max-min problem can then be stated as follows:

$$\max t \quad (1)$$

$$\text{s.t.} \quad t \leq f_j, \quad j \in J \quad (2)$$

$$\sum_{p \in M_j} f_{jp} = f_j, \quad j \in J \quad (3)$$

$$\sum_{j \in J} \sum_{p \in M_j} \delta_{ejp} f_{jp} \leq c_e, \quad e \in E \quad (4)$$

$$f_{jp} \geq 0, \quad j \in J, p \in M_j \quad (5)$$

Equation (3) aggregates bifurcated flows for each service, while (4) prevents the flows from exceeding the link capacity. Optimization problem (1)–(5) is easy to solve if the sets of paths M_j are predefined and small. If, however, all feasible paths are allowed to make up sets M_j , the number of columns can increase exponentially with the size of the graph.

The solution is to use the column (path) generation technique [4, 11], where not all the columns of the constraints matrix are stored. Instead, only a subset of the variables (columns) that can be seen as an approximation (restriction) of the original problem is kept. The column generation algorithm iteratively modifies the subset of variables by introducing new variables in a way that improves the current optimal solution. At the end, the set contains all the variables (paths) necessary to construct the overall optimal solution which can use all possible paths in the graph. New columns are generated in the *pricing problem*.

Let $(\lambda_j)_{j \in J}$ and $(\pi_e)_{e \in E}$ be current optimal dual variables associated with constraints (3) and (4), accordingly. At each iteration we are interested in generating

path p for which the reduced price $\sum_{e \in E} \pi_e \delta_{ejp} + \lambda_j$ has the smallest and negative value, as we can expect this will improve the current optimal solution as much as possible. It is easy to note that pricing problem corresponds to finding the shortest path connecting two nodes in the graph with the link lengths determined by the dual vector π . The algorithm stops when there are no more paths for which the reduced price is negative.

3 Fair Aggregation Operators

As stated before the basic operator used to preserve fairness among outcomes is max-min, regularized by lexicographic maximization of the second worst outcomes (provided that the worst one remains as large as possible), third worst outcomes (provided that the two largest remain as large as possible), and so on (MMF). In the case of linear problem it is possible to carry out the MMF procedure based on simple algorithm that in each step uses the dual information to determine the outcomes that are blocked at their highest values possible. In the following steps, only the outcomes are optimized that have not been blocked before (for details see [11]).

The distribution based aggregation operators presented in this paper allow more flexibility in expressing decision maker preferences with respect to, both, total throughput and fairness of the solution. In the OWA aggregation of outcomes $\mathbf{y} = (y_1, \dots, y_m)$ weights $\mathbf{w} = (w_1, w_2, \dots, w_m)$ are assigned to the ordered values rather than to the specific criteria:

$$A_w = \sum_{i=1}^m w_i \theta_i(\mathbf{y}) \tag{6}$$

where $(\theta_1(\mathbf{y}), \theta_2(\mathbf{y}), \dots, \theta_m(\mathbf{y})) = \Theta(\mathbf{y})$ is the ordering map $R^m \rightarrow R^m$ with $\theta_1(\mathbf{y}) \leq \theta_2(\mathbf{y}) \leq \dots \leq \theta_m(\mathbf{y})$ and there exists a permutation τ of set I such that $\theta_i(\mathbf{y}) = y_{\tau(i)}$ for $i = 1, 2, \dots, m$.

If the weights are monotonic $w_1 > w_2 > \dots > w_{m-1} > w_m$, the OWA aggregation has the property of equitability [9], that guarantees that an equitable transfer of an arbitrarily small amount from the larger outcome to a smaller outcome results in more preferred achievement vector. Every solution maximizing the OWA function is then an equitably efficient solution to the original multiple criteria problem. Moreover, for linear multiple criteria problems every equitably efficient solution can be found as an optimal solution to the OWA aggregation with appropriate weights.

For the maximization problem the OWA objective aggregation can be formulated as linear extension of the original problem, as follows. Let us apply linear cumulative map to the ordered achievement vectors $\Theta(\mathbf{y})$

$$\bar{\theta}_k(\mathbf{y}) = \sum_{i=1}^k \theta_i(\mathbf{y}) \quad k = 1, 2, \dots, m \tag{7}$$

As stated in [9], for any given vector $\mathbf{y} \in R^m$, the cumulated ordered coefficient $\bar{\theta}_k(\mathbf{y})$ can be found as the optimal value of the following LP problem:

$$\bar{\theta}_k(\mathbf{y}) = \max kt_k - \sum_{i=1}^m d_{ki} \tag{8}$$

$$\text{s.t. } t_k - y_i \leq d_{ki}, d_{ki} \geq 0 \quad i = 1, 2, \dots, m \tag{9}$$

The ordered outcomes can be expressed as differences $\theta_i(\mathbf{y}) = \bar{\theta}_i(\mathbf{y}) - \bar{\theta}_{i-1}(\mathbf{y})$ for $i = 2, \dots, m$ and $\theta_1(\mathbf{y}) = \bar{\theta}_1(\mathbf{y})$. Hence, the maximization of the OWA operator [6] with weights w_i can be expressed in the form:

$$\max \left\{ \sum_{i=1}^m w'_i \bar{\theta}_i(\mathbf{y}) : \mathbf{y} \in Y \right\} \tag{10}$$

where coefficients w'_i are defined as $w'_m = w_m$ and $w'_i = w_i - w_{i+1}$ for $i = 1, 2, \dots, m - 1$ and Y is the feasible set of outcome vectors \mathbf{y} . If the original weights w_i are strictly decreasing, then $w'_i > 0$ for $i = 1, 2, \dots, m$.

For the multiple commodity flow optimization problem [1]–[5] the final OWA aggregation of the outcomes f_j for all services can be stated as the following LP model:

$$\max \sum_{k=1}^{|J|} kw'_k t_k - \sum_{k=1}^{|J|} \sum_{j \in J} w'_k d_{jk} \tag{11}$$

$$\text{s.t. } d_{jk} \geq t_k - f_j, d_{jk} \geq 0 \quad k = 1, 2, \dots, |J|, j \in J \tag{12}$$

$$\mathbf{f} \in F \tag{13}$$

where $\mathbf{f} = [f_j]_{j \in J}$ and F is a feasible set of flows/throughputs defined by [3]–[5].

The WOWA aggregation is OWA generalization that allows assigning importance weights to specific criteria [8]. Those weights could express, for example, relative importance of the services. The weights assigned to ordered values will be further called preferential weights.

Let $\mathbf{s} = (s_1, \dots, s_m)$ be an m -dimensional vector of importance weights such that $s_i \geq 0$ for $i = 1, \dots, m$ and $\sum_{i=1}^m s_i = 1$. The corresponding Weighted OWA aggregation of vector \mathbf{y} is defined [13] as follows

$$A_{w,s} = \sum_{i=1}^m \omega_i \theta_i(\mathbf{y}) \tag{14}$$

with

$$\omega_i = w^* \left(\sum_{k \leq i} s_{\tau(k)} \right) - w^* \left(\sum_{k < i} s_{\tau(k)} \right), \tag{15}$$

where w^* is an increasing function interpolating points $(i/m, \sum_{k \leq i} w_k)$ together with the point $(0, 0)$ and τ representing the ordering permutation for \mathbf{y} (i.e. $y_{\tau(i)} = \theta(\mathbf{y})$).

Moreover, function w^* is required to be a straight line when the points can be interpolated in this way. We assume the piecewise linear interpolation function w^* which is the simplest form of the required interpolation.

Note, that the piecewise linear functions may be built with various number of breakpoints, not necessarily m [8]. Thus, any nonlinear function can be well approximated by a piecewise linear function with appropriate number of breakpoints. Therefore, we will consider weights vectors \mathbf{w} of dimension n not necessarily equal to m . It is even possible to define a generalized WOWA aggregation where the preferential weights w_k are allocated to an arbitrarily defined grid of ordered outcomes defined by quantile breakpoints (see [8] and references therein).

As shown in [8], maximization of an equitable WOWA aggregation with decreasing preferential weights $w_1 \geq w_2 \geq \dots \geq w_n$ may be implemented as the LP expansion of the original problem. In the case of the multiple commodity flow optimization problem (3)–(5), this can be stated as follows:

$$\max \sum_{k=1}^n w'_k \left[\frac{k}{n} t_k - \sum_{j \in J} s_j d_{jk} \right] \tag{16}$$

$$\text{s.t. } d_{jk} \geq t_k - f_j, d_{jk} \geq 0 \quad k = 1, 2, \dots, n, j \in J \tag{17}$$

$$\mathbf{f} \in F \tag{18}$$

If the importance weights are equal $s_j = 1/|J|$, the model reduces to the OWA aggregation.

A special case of the generalized WOWA aggregation is defined for single breakpoint and corresponds to optimization of the predefined quantile of worst outcomes and in finance is known as the CVaR (Conditional Value at Risk). It can be computed as a standard linear extension of the original problem [8]:

$$\max t - 1/\beta \sum_{j \in J} s_j d_j \tag{19}$$

$$\text{s.t. } d_j \geq t - f_j, d_j \geq 0 \quad j \in J \tag{20}$$

$$\mathbf{f} \in F \tag{21}$$

4 Numerical Experiments

We analyzed the performance of the CVaR and WOWA aggregation operators together with the performance of the classic max-min and lexicographic max-min (MMF) operators. The aggregation operators were applied to the problem defined by the constraints (3)–(5) with the network flows f_j as the optimization criteria.

For the experiments we used a set of 10 randomly generated problems for each tested size. The problem are based on randomly generated networks, for which random node pairs were chosen to define services.

Although weights determination for WOWA is an important issue in the theory of Ordered Weighted Averaging [14, 1, 2], for the performance check simple

Table 1 Computing times [s]

Problem size			Aggregation operator					
$ J $	$ E $	$ V $	Max-Min	MMF	CVaR	WOWA	WOWA2	
50	100	80	0.2	0.3	0.0	1.2	0.3	
50	200	80	0.3	1.0	0.1	5.3	1.6	
50	400	80	0.6	4.3	0.6	19.4	6.9	
50	800	80	1.6	25.2	3.1	67.4	30.6	
100	100	80	0.4	0.6	0.1	39.1	6.4	
100	200	80	0.5	1.5	0.1	219.9	34.3	
100	400	80	1.0	4.2	1.0	353.4	74.8	
100	800	80	3.2	17.3	6.4	578.1 ³	166.7	
200	100	80	0.9	1.2	0.1	–	257.4	
200	200	80	1.0	2.5	0.2	–	–	
200	400	80	1.8	5.7	1.9	–	–	
200	800	80	4.7	17.7	8.3	–	–	

Table 2 Number of columns generated

Problem size			Aggregation operator					
$ J $	$ E $	$ V $	Max-Min	MMF	CVaR	WOWA	WOWA2	
50	100	80	54.1	65.0	9.9	7.1	6.8	
50	200	80	67.5	103.9	18.3	26.0	26.4	
50	400	80	127.9	177.6	32.9	52.0	52.8	
50	800	80	213.6	303.6	56.5	95.2	94.4	
100	100	80	104.1	118.1	10.1	7.0	6.8	
100	200	80	117.7	145.6	13.9	19.8	20.3	
100	400	80	170.7	208.5	27.3	33.5	33.3	
100	800	80	266.5	312.7	50.5	51.9 ³	52.1	
200	100	80	203.1	219.1	9.3	–	6.8	
200	200	80	215.0	240.4	13.4	–	–	
200	400	80	261.7	287.3	23.9	–	–	
200	800	80	366.9	392.8	37.6	–	–	

generation methodology has been chosen. All the weights, except two, are strictly decreasing numbers with the step 0.1, while the two selected weights ($k = \lfloor n/3 \rfloor$ and $k = \lfloor 2n/3 \rfloor$) differ from the previous ones by 0.5. The importance weights were generated as random numbers in the range 0.5 to 1.0 and then normalized.

All the experiments were performed on the Intel Core i7 2.9GHz microprocessor using CPLEX 12.1 optimization library for the linear master problem. The results are the average of 10 randomly generated problems of a given size. Computing times are presented in Table 1 and the total number of the columns generated by the algorithm in Table 2. The index denotes the number of tests (out of 10) that exceeded the maximum computation time of 600 seconds and the – sign means that all tests ended up with timeout. The WOWA2 column shows the results for the case with the

reduced (by factor 2) number of importance weights (the weights were allocated to an uniform grid of quantile breakpoints).

One can notice the advanced aggregation operators generally have longer computing times than MMF, but much lower number of iterations. The reason is rather high number of additional linear constraints ($O(n|J|)$) of the WOWA operator. This becomes more evident when looking at the results for the CVaR, where the number of additional constraints is of order $O(|J|)$ or for the WOWA2 with reduced number of preferential weights n . This suggests better results in cases with more difficult pricing problem, where the number of iterations plays important role.

5 Conclusion

The problem of fair resource allocation is of considerable importance in multi-commodity network-based market systems. Advanced aggregation operators based on the Ordered Weighted Averaging allow to model diverse preferences with regard to fairness and efficiency. We have shown that application of the advanced aggregation operators for the multiple commodity flow optimization problems requires nothing more than a number of additional linear constraints and in some cases or using some approximations can be also very effective. The generality of the introduced models makes them easy to apply to a variety of computer systems engineered for multi-commodity trading markets.

Acknowledgements. The research was supported by the Polish National Budget Funds 2010-2013 for science under the grant N N514 044438.

References

1. Ahn, B.S.: Preference relation approach for obtaining OWA operators weights. *Int. J. Approx. Reason.* 47, 166–178 (2008)
2. Amin, G.R.: Notes on properties of the OWA weights determination model. *Comput. Ind. Eng.* 52, 533–538 (2007)
3. Bell, D.E., Raiffa, H.: Risky choice revisited. In: Bell, et al. (eds.) *Bell at all, Decision Making Descriptive, Normative and Prescriptive Interactions*, pp. 99–112. Cambridge University Press, Cambridge (1988)
4. Dantzig, G.B., Wolfe, P.: The decomposition algorithm for linear programming. *Oper. Res.* 8, 101–111 (1960)
5. Kaleta, M., Traczyk, T. (eds.): *Modeling Multi-commodity Trade*. AISC, vol. 121. Springer, Heidelberg (2012)
6. Kostreva, M.M., Ogryczak, W.: Linear optimization with multiple equitable criteria. *RAIRO Operations Research* 33, 275–297 (1999)
7. Ogryczak, W.: Multiple criteria linear programming model for portfolio selection. *Ann. Oper. Res.* 97, 143–162 (2000)
8. Ogryczak, W., Śliwiński, T.: On Efficient WOWA Optimization for Decision Support under Risk. *International Journal of Approximate Reasoning* 50, 915–928 (2009)

9. Ogryczak, W., Śliwiński, T.: On Solving Linear Programs with the Ordered Weighted Averaging Objective. *European Journal of Operational Research* 148, 80–91 (2003)
10. Ogryczak, W., Śliwiński, T.: On Optimization of the Importance Weighted OWA Aggregation of Multiple Criteria. In: Gervasi, O., Gavrilova, M.L. (eds.) *ICCSA 2007, Part I*. LNCS, vol. 4705, pp. 804–817. Springer, Heidelberg (2007)
11. Pióro, M., Medhi, D.: *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufman, San Francisco (2004)
12. Rawls, J.: *The Theory of Justice*. Harvard Univ. Press, Cambridge (1971)
13. Torra, V.: The weighted OWA operator. *Int. J. Intell. Syst.* 12, 153–166 (1997)
14. Wang, Y.M., Parkan, C.: Minimax disparity approach for obtaining OWA operator weights. *Info. Sci.* 175, 20–29 (2005)
15. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Systems, Man and Cybernetics* 18, 183–190 (1988)
16. Yager, R.R., Kacprzyk, J. (eds.): *The Ordered Weighted Averaging Operators, Theory and Applications*. Kluwer Academic Publisher, Boston (1997)
17. Yager, R.R., Filev, D.P.: *Essentials of Fuzzy Modeling and Control*. Wiley, New York (1994)

Part V
Social Data Processing

Heuristic Approach to Automatic Wrapper Generation for Social Media Websites

Bartosz Baziński and Michał Brzezicki

Abstract. The data contained within user generated content websites prove to be valuable in many applications, for example in social media monitoring or in acquisition of training sets for machine learning algorithms. Mining such data is especially difficult in case of web forums, because of hundreds of various forum engines used. We propose an algorithm capable of unsupervised extraction of posts from social websites without the need to analyse more than one page in advance. Our method localizes potential data regions by repetition analysis within document structure and filtering potential results. Subsequently the fields of data records are found using key characteristics and series-wide dependencies. We managed to achieve 87% precision of extraction and 82% recall after experiments on single pages taken from 231 websites. Our solution is characterized by high computing efficiency, thus enabling wide applications.

Keywords: automatic wrapper generation, information extraction, social media websites, web forums.

1 Introduction

1.1 Problem Description

Never in the history of mankind so much data has been recorded and stored in accessible way. The estimated size of the world wide web reaches 50 billion

Bartosz Baziński · Michał Brzezicki
Faculty of Electronics, Telecommunications and Informatics,
Gdansk University of Technology, 80-233 Gdańsk, Poland
e-mail: bartek@bazinski.com, mbrzezicki@gmail.com
<http://www.thivera.pl>

web pages¹. Although data source is almost infinite, information is provided in very incoherent way making it hard to gather in unified database. Vast majority of data is scattered among numerous blogs, forums, news sites, etc. Automated extraction of the data from such sites is a difficult task, because each source has different structure of HTML tags.

A program capable of performing such extraction is called wrapper. Wrapper is defined as a procedure that translates content from specific information source into a relational model [8]. In case of extraction from social media websites, we would define input to the wrapper as a text string containing HTML structure of a web page and output as a list of statements contained within that page, with content, author and date of creation for each statement. Although creation of such wrapper for a single source is an relatively easy task and can be performed manually, problems arise when we need to obtain data from thousands of diverse websites. With almost every site using different layout template, we require a tool capable of creating wrappers, entitled wrapper generator.

1.2 Applications of Wrapper Generators

Development of an efficient wrapper generator - both in terms of computing performance and extraction precision - would have practical applications in numerous cases. Having wrappers for web forums would allow access to enormous amount of discussions on various topics. Those posts could be mined and used to provide direct answers to questions of users in Q&A websites and Internet search engines [2].

Another field of high application potential is social media monitoring. With the explosion of Web 2.0 platforms such as blogs and discussion forums, the consumers have gained an unimaginable abilities to share their opinions with others and influence their decisions. Knowing what consumers say can enable companies to derive valuable marketing intelligence [5] and better align all their activities within needs of their clients [15]. Due to increasing fragmentation of media, the traditional, manual methods of observation and data gathering were rendered useless [7]. In order to automatically oversee conversations in social media, one must implement effective extraction of content. Web forums, blogs and review sites are only accessible through web crawling, hence comes the need for wrapper generator. For precise data analysis, the author, date, statement and context of a post is needed. Statements extracted from HTML page structure could be further used, for example in sentiment detection.

Other example use of content extraction would be in preparation of training sets for text-analysis algorithms. Many websites contain posts already categorized by some criterion. For example product reviews site with

¹ The size of the World Wide Web, <http://www.worldwidewebsize.com>

opinions divided by emotional affection of the author could be mined and used as a teaching set for sentiment classifier.

1.3 Related Work

With the growing popularity of the Internet, more and more scholars are engaged in research on wrappers generators. Wrapper induction is one of the main approaches to the problem. Manually labelled teaching set is used to derive extraction rules during supervised learning. There are few notable works in this field. The first one is STALKER algorithm [14], which generates formal data extraction rules for selected websites. It requires only few training examples and uses a set of disjunctive landmark automata organized in a hierarchy. The second often referenced technique is Boosted Wrapper Induction [4]. It uses machine learning algorithm improved by repeatedly applying it to the training set with different example weightings.

The requirement of manual work is a considerable disadvantage of mentioned methods. Drawbacks would be especially visible in a long-term continuous data extraction (for example in social media monitoring). When the structure of the source changes, the initial set preparation and teaching process would have to be performed again. Due to this aspects some research has been made on automation of wrapper maintenance. The proposed machine learning approach [10] is based on an idea of storing the correctly extracted data chunks. In case of a source template change the training set is able to self-update the position of labels by searching for content matching previously acquired information.

Nevertheless, the shortcomings of wrapper induction motivated researchers to pursue more universal generation methods. The idea behind the automatic wrapper generation approach is to extract data by searching for patterns in the page content. Additional heuristics are used to help locate data series and data fields inside each data record. Of course information from various domains has different characteristics, which makes it difficult to establish universal algorithms.

There are substantially more solutions in this category. ROADRUNNER [3] is based on analysis of similarities in documents and creates a wrapper by inferring a generalized regular expression after analysing multiple pages. IEPAD [1] takes into consideration that most of the websites with user generated content uses only one template for displaying the data from one tuple and automatically identifies records by repeated pattern mining of HTML structures. More adaptable ODE [19] algorithm harnesses domain specific ontologies to identify data regions and to align and label the data values in records. DEPTA [22] renders analysed pages and uses visual boundaries of data records together with string edit distance to locate data series. There are also methods of statistical origin, which do not require any domain specific

knowledge or heuristics. Hierarchical clustering algorithms could be used for data region discovery and segmentation [16]. Another approach could be to analyse source using Markov Logic Networks in order to efficiently perform distinction between parts of the document belonging to layout and those being data record [17].

Some algorithms make use of the multi-level examination that takes into account that often lists contain links to single pages, where each record is presented in details [9]. Such knowledge improves the record localization. However, this method has no use in analysis of social media, where only few websites have special detailed page for each statement.

Approaches dedicated to extracting data from web forums include:

- Analysis of repetitions of both tag structures and visible text tokens [12].
- Using specific domain constraints and utilising them for data region segmentation [18].
- Site-wide analysis of multiple pages in terms of fulfilling certain features [21]. Markov model networks are used to build probabilistic model for localising data records and their data fields.

Nevertheless, the field of wrapper generation is still open to new ideas. Many analyses show that aforementioned techniques are not sufficient [13, 20].

2 Algorithm

2.1 Motivation

The lack of efficient techniques for generic statement extraction was our primary issue during collection of data from social media websites. In result, we have decided to design and implement a wrapper generation algorithm for the user generated content websites. The main goals were to achieve high precision and recall, ensure complete automation of wrapper generation that would result in no manual maintenance work. We also wanted to establish high computing efficiency that would allow us to use the algorithm in distributed web crawling system and mine posts from c.a. 5 thousands websites.

2.2 General Idea of the Algorithm

The developed algorithm is based on the analysis of the document tree by creating list of all potential candidate data regions, then filtering and scoring is applied to select one data region that most probably contains posts from users. Afterwards, the final region is analysed by various heuristics to locate

data fields inside each data record. The overall procedure of the algorithm could be stated as follows:

1. Convert HTML mark-up of the document to a tree object model.
2. Create initial data regions set by searching for repetitive tag names.
3. Select final data region and data records by filtering using the following methods:
 - (a) Tag Number Filter
 - (b) Dummy Tree Matching Filter
 - (c) Data Exists Filter
 - (d) Score Filter
4. Locate most common paths for data fields inside the data region:
 - (a) Locate date of post path using Date Matching Location Finder
 - (b) Locate content path using Edit Distance Location Finder

The initial data regions set creation and Dummy Tree Matching Filter are based on the methods proposed in the Tree Wrap algorithm [6].

2.3 HTML Mark-Up to Tree Object Model Conversion

HTML4 is not an XML-compliant standard, in consequence requiring conversion to acquire proper tree-based document model. The newer XHTML standard is theoretically compatible with XML. However, in reality many of the webpages fail the W3C mark-up validation test. This results in the need to clean up the HTML code to correct minor errors, e.g. unclosed tags, tag names misspellings, etc. Additionally, at this step we remove 34 irrelevant text-formatting tags. This causes the flattening of the document tree model and eases the creation of wrapper.

2.4 Initial Data Regions Set Creation

The main observation is that data records are contained within nodes with equivalent tag names. Almost invariably those nodes are on corresponding level, having the same root node. Using these findings we create the initial candidate set by searching for repetitive tags on each possible level. Search begins at the very root node and recursively delves into each child. Every possible tag name that occurs 3 or more times as a child of the examined node is taken into account as a potential data region. At the end of this procedure we obtain a list of all candidate data regions.

2.5 *Filtering Data Regions*

The second step consists of applying various filters that remove irrelevant data regions and/or data records. After applying each filtering step, all data regions containing less than 3 data records are removed.

2.5.1 **Tag Number Filter**

This filter takes advantage of the observation that each post (i.e. data record) has relatively complex HTML code and it should contain at least 3 data fields (content, author, date). It counts number of all nodes inside tree structure of each data record, if it contains less than 3 nodes, the data record is removed from the data region.

2.5.2 **Dummy Tree Matching Filter**

In this step we take into consideration the fact that websites use templates for code generation. In consequence each of the data records should follow similar tag structure. Only minor differences should be noted, due to the use of formatting tags in the content of the data record. The filter matches the tree structures of records and removes those with anomalies. Matching is performed in a "dummy" way, by counting the number of unique tags in each data record at the top level and with recursion at all levels. If the difference exceeds the threshold, record is removed.

2.5.3 **Date Exists Filter**

This phase derives from a conclusion that each of the data records must contain a date of post creation in some text format. During collection of 231 test websites we were not able to find any website that would not display the dates of posts. However, checking whether the examined data record has a date field turned out to be exceptionally difficult task. Many of web forums engines allow to customize the date display format. This results in extremely wide range of possible date formats. Additional problem appeared because majority of our test cases were Polish web forums. We had to include specific matchers for Polish language that were able to recognize months in various linguistic forms. For instance January can be written in Polish as "Styczeń", "Styczniu" or "Stycznia" depending on the day of the month. Moreover, we also had to implement recognition for various common cases:

- misconfiguration of the date display format, often resulting in doubled year ("17 lutego 2012, 2012 19:32") or AM/PM mark in 24-hour clock ("8 Jan 2009, 17:45 pm"),

- usage of the "<time>ago" patterns (e.g. "3 minutes 20 seconds ago", "1 year 7 days ago", etc.),
- displaying day and months numbers using Roman numerals ("22 II 2012, 17:53"),
- different combinations of punctuation marks ("20.Jun.2011 14:53"),
- character strings looking similar to dates ("Version 3.12.8").

The resulting date string recognition tool consists of 61 regular expression matchers, divided into 3 groups. Each group have different text preprocessing that removes characters irrelevant to the matcher group. We prepared a test suite containing 307 tests for various date formats, including 22 tests for existence of false positives.

2.5.4 Score Filter

The final step of filtering involves scoring resulting data regions. Only region with highest score is passed to the next phase of algorithm. This step is based on the following observations:

1. Regions containing posts are relatively large comparing to the whole page (text length multiplier).
2. User generated content is usually deeply embedded in HTML structure (html depth multiplier).

The scoring function equation is as following:

$$SCORE(dataregion) = length * L + depth * D \quad (1)$$

Where: *length* - text length of the data region; *L* - text length multiplier (constant); *depth* - level of depth of the data region in HTML tree structure; *D* - depth multiplier (constant). Exact parameters for the scoring function can be found using genetic search algorithms, depending on the test set.

2.6 Locating Fields Inside Data Records

The previous stage of the algorithm gives us a list of parts of website that may be statements - the data records within the final data region. From those records we extract final information - for each post we retrieve content, date and author. For each of the fields above we seek location that is common for all given records. Note that any found location can be represented as an XPath expression.

2.6.1 Content Location

Analysing HTML source we can make observation that content is always in the same location but the inner text of it is very variable. Therefore, we perform edit distance measurements between possible content locations with the idea, that the location with biggest differences between records would be our content location. More exactly, the algorithm works as follows:

1. For every data record we look for node with longest text content. We save to the candidates list a generic XPath expression allowing to localise that node.
2. After searching all data records, we remove duplicated location expressions.
3. We calculate the score of each location expression by applying it to each data record. Resulting series of possible post contents is examined using the Levenstein edit distance algorithm [11] to measure differences between each other.
4. The expression that localises the series with the biggest differences between each content text is returned as the common location of content nodes.

2.6.2 Date Location

Every node in record which is short enough is matched against generic date recognition tool, that was described earlier in Date Exists Filter. Note that data record can contain many dates (i.e. when author joined forum, when post was written, date of last edition) but only the most recent one is returned. Similar to the previous step, we find the most common location of date in data record and return it.

2.6.3 Author Location

This step is optional, as not every statement has author or it can be hidden in content. Nevertheless, if there is node which HTML class attribute contain "author" and its content is short enough we may suppose it contains name of the author. Like in previous step we return the most common location of author node in data record.

3 Experiments

3.1 The Test Set

We have manually prepared a test set containing 231 HTML pages in Polish and English language from web forums and other social media websites. Table

Table 1 Web forums engines statistics

Engine name (various versions included)	Count
phpBB	66
Custom engines	65
IP.Board	42
vBulletin	31
Other open-source engines	27

1 contains statistics of the forum engines used in the tests pages. All pages contain a sum of 2805 posts.

3.2 Methodology

For every web page there is a meta document describing expected extraction result. Every meta document includes two "golden posts": one from the beginning of a web page and the other from the end. There are only two posts, because if page is extracted correctly, then all posts are correct and none otherwise. Nevertheless, there are some minor errors which are acceptable for us. These errors are:

1. Difference in expected and actual post count by 1 - usually first, last or post that contains advertisement.
2. No hour and minute in extracted date.
3. Extracted post does not contain the whole actual post because of complexity (quotations or other blocks).

Therefore there are two test suites - one which accepts the minor errors (tolerant test) and second that does not allow them (strict test). There are also two cases that partially test our algorithm. First is the test of extracting records. It checks initial data region creation and filtering of data records. Second test focuses only on locating fields inside data records. Both of these tests are strict tests and does not allow any errors. We have computed the following widely used indicators:

- Recall - number of correctly extracted documents/number of expected documents.
- Precision - number of correctly extracted documents/number of extracted documents.
- F-measure - $2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$.
- Correctness - number of passed test cases/all test cases.

3.3 Test Results

Table 2 Results of the application of the algorithm on social websites

Test type	Strict algorithm	full	Tolerant algorithm	full	Extracting data records	Locating fields inside data records
Test cases	231		231		228	228
Recall	72.76%		82.45%		78.53%	82.02%
Precision	76.67%		86.88%		81.80%	82.88%
F-measure	0.7466		0.8461		0.8013	0.8245
Correctness	76.19%		82.68%		78.53%	85.52 %

4 Summary

We find our results very promising on the account that our algorithm requires only 1 webpage to generate wrapper and is fully automatic. We believe that with the small errors tolerance the algorithm correctness is sufficient to allow integration in production systems. Therefore we have run our wrapper generation algorithm on a 4-server crawling cluster. Each of the computers was based on an Intel Core i5 2.66GHz processor with 16GB RAM. The 24-hour test on 2197 social media sites allowed us to generate an average of 145,16 wrappers per second, in an environment where connection bandwidth was not the bottleneck.

We plan to address the imperfections of the algorithm by broadening the analysis to multiple pages from each website. Such approach would eliminate the errors resulting from single-page anomalies. For example, discussion with very short answers would render the content location finder useless. Other case would be a series of posts with big differences in structure of content. Such occurrence could result in Dummy Tree Matching Filter incorrectly removing data records.

In summary, we evaluate the algorithm as very useful. We are currently using it to extract posts from forums on capital markets. We intend to analyse the sentiment of the statements and seek correlation with stock market behaviour.

Acknowledgements. Presented work has been done as a part of Thivera research project, supported by the Foundation for Polish Science [Ventures/2011-7/1].

References

1. Chang, C.-H., Lui, S.-C.: IEPAD: information extraction based on pattern discovery. In: Proceedings of the 10th International Conference on World Wide Web, pp. 681–688. ACM, Hong Kong (2001)

2. Cong, G., et al.: Finding question-answer pairs from online forums. In: SIGIR 2008 Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 467–474. ACM, Singapore (2008)
3. Crescenzi, V., Mecca, G., Merialdo, P.: RoadRunner: automatic data extraction from data-intensive web sites. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. ACM, Madison (2002)
4. 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 (2000)
5. Glance, N., et al.: Deriving marketing intelligence from online discussion. In: KDD 2005 Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 419–428. ACM, Chicago (2005)
6. Hong, J.L., Fauzi, F.: Tree Wrap-data Extraction Using Tree Matching Algorithm. *Majlesi Journal of Electrical Engineering* 4(2) (2010)
7. Kim, P.: The forrester wave: Brand monitoring, Q3 2006, Forrester Wave (2006) (white paper)
8. Kushmerick, N., Weld, D., Doorenbos, R.: Wrapper Induction for Information Extraction. In: Proceedings of the International Joint Conference on Artificial Intelligence (1997)
9. Lerman, K., Getoor, L., Minton, S., Knoblock, C.: Using the structure of Web sites for automatic segmentation of tables. In: SIGMOD 2004 Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, pp. 119–130. ACM, Paris (2004)
10. Lerman, K., Minton, S.N., Knoblock, C.A.: Wrapper maintenance: a machine learning approach. *Journal of Artificial Intelligence Research* 18(1), 149–181 (2003)
11. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics - Doklady* 10(8), 707–710 (1966)
12. Li, S., Tang, L., Hu, J., Chen, Z.: Automatic Data Extraction from Web Discussion Forums. In: FCST 2009 Proceedings of the 2009 Fourth International Conference on Frontier of Computer Science and Technology, pp. 219–225. IEEE Computer Society Press, Brak Miejsca (2009)
13. Liu, B., Grossman, R., Zhai, Y.: Mining data records in Web pages. In: KDD 2003 Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 601–606. ACM, Washington, DC (2003)
14. Muslea, I., Minton, S., Knoblock, C.: STALKER: Learning Extraction Rules for Semistructured. In: Web-based Information Sources, AAAI (1998)
15. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2(1-2) (2008)
16. Papadakis, N., Skoutas, D., Raftopoulos, K., Varvarigou, T.: An Automatic Web Wrapper for Extracting Information from Web Sources, Using Clustering Techniques. In: SAINT 2005 Proceedings of the 2005 Symposium on Applications and the Internet, pp. 24–30. IEEE Computer Society, Washington, DC (2005)
17. Satpal, S., et al.: Web information extraction using markov logic networks. In: KDD 2011 Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1406–1414. ACM, San Diego (2011)

18. Song, X., et al.: Automatic extraction of web data records containing user-generated content. In: CIKM 2010 Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 39–48. ACM, Toronto (2010)
19. Su, W., Wang, J., Lochovsky, F.H.: ODE: Ontology-assisted data extraction. *ACM Transactions on Database Systems* 34(2) (2009)
20. Weninger, T., et al.: Unexpected results in automatic list extraction on the web. *ACM SIGKDD Explorations Newsletter* 12(2), 26–30 (2011)
21. Yang, J.-M., et al.: Incorporating site-level knowledge to extract structured data from web forums. In: WWW 2009 Proceedings of the 18th International Conference on World Wide Web, pp. 181–190. ACM, Madrid (2009)
22. Zhai, Y., Liu, B.: Web data extraction based on partial tree alignment. In: WWW 2005 Proceedings of the 14th International Conference on World Wide Web, pp. 76–85. ACM, Chiba (2005)

Spectral Clustering: Left-Right-Oscillate Algorithm for Detecting Communities

Pavla Dráždilová, Jan Martinovič, and Kateřina Slaninová

Abstract. Detection of communities in the complex networks is an actual problem solved in research area. The paper describes a new algorithm for this purpose. Left-Right-Oscillate algorithm (LRO) is based on spectral ordering of graph vertices. This approach allows us to detect a desired community – either by the size of the smallest communities or by the level of modularity. Since the LRO algorithm detects efficiently communities in large network even when these are not sharply partitioned, it turns to be specially suitable for the analysis of social, complex or coauthor networks. In this paper, proposed algorithm is used for finding communities in a large coauthor network - DBLP.

1 Introduction

The important property of many networks is a community structure, the division of network nodes into groups within which the network connections are dense, but between which they are sparser [22].

Detecting the community structure in complex networks allows us to obtain information in a more efficient way. When used in the analysis of large collaboration networks (for example companies or universities), communities reveal the informal organization and the nature of information flows through the whole system [27]. For searching communities in a large network, we based our work on the method of spectral clustering to efficiently find clusters (communities) of authors that published together.

The main goal of the paper is the description of our proposed Left-Right-Oscillate algorithm for the community detection in large networks. The core of this work is

Pavla Dráždilová · Jan Martinovič · Kateřina Slaninová
VŠB - Technical University of Ostrava,
Faculty of Electrical Engineering and Computer Science,
17. Listopadu 15/2172, 708 33 Ostrava, Czech Republic
e-mail: [pavla.drazdilova,jan.martinovic}@vsb.cz](mailto:{pavla.drazdilova,jan.martinovic}@vsb.cz),
katerina.slaninova@vsb.cz

based on spectral ordering. Spectral ordering is the first part of an algorithm used to seek out communities within network. More precise designations for communities are then monitored using modularity [20].

2 Social Network

Social networking is a complex, large and expanding sector of the information economy. Researchers' interest in this field is growing rapidly. It has been studied extensively since the beginning of the 20th century. The first normative contributions in this area were proposed in 1970s by sociologist Mark Granovetter and mathematician Linton C. Freeman. The basic theory "The Strength of Weak Ties" was mentioned in 1973 [13]. Another significant principle was published in 1979 by Linton C. Freeman [10]. In his work was presented definition of centrality, which is one node's relationship to other nodes in the network. He defined basic metrics like degree, control and independence, from which reason researchers proceed in their present works.

Social network (SN) is a set of people or groups of people with similar patterns of contacts or interactions such as friendship, co-working, or information exchange [11]. The World Wide Web, citation networks, human activity on the internet (email exchange, consumer behavior in e-commerce), physical and biochemical networks are some examples of social networks. Social networks are usually represented by graphs, where nodes represent individuals or groups and lines represent contacts among them. The configuration of relations among network members identifies a specific network structure. This structure can vary from isolated structures where no members are connected to saturated structures in which everyone is interconnected.

In our work, we used Gephi¹ for the visualization of the network structure. Gephi [1] is an open-source network analysis and visualization software package written in Java on the Netbeans platform. Gephi has been selected for the Google Summer of Code in 2009, 2010 and 2011. Gephi has been used in a number of research projects in the university or journalism sphere, for instance for visualizing the global connectivity of New York Times content [15] or mapping dynamic conversation networks on Twitter [2].

2.1 Community Detection

Discovery and analysis of community structure in networks is a topic of considerable recent interest in sociology, physics, biology and other fields. Networks are very useful as a foundation for the mathematical representation of a variety of complex systems such as biological and social systems, the Internet, the world wide web, and many others [17, 8, 21]. A common feature of many networks is their community

¹ <http://gephi.org/>

structure, the tendency for vertices to divide into groups, with dense connections within groups and only sparser connections between them [12, 18]. Social networks [12] and information networks such as the web [9] have all been shown to possess strong community structure; finding that has substantial practical implications for our understanding of the systems these networks represent. Newman and Girvan [22] proposed algorithms for finding and evaluating community structure in network. They used a divisive technique which iteratively removes edges from the network, thereby breaking it up in communities. The edges to be removed are identified by using one of a set of edge betweenness measures, of which the simplest is a generalization to edges of the standard shortest-path betweenness of Freeman. Then, their algorithms include a recalculation step in which betweenness scores are reevaluated after the removal of every edge.

To detect communities, graph partitioning methods or hierarchical clustering has been applied. Originally, graph partitioning methods, based on edge removal [24], divide the vertices of a network into a given number of (non-overlapping) groups of a given size, while the number of edges between groups is minimal.

Newman used eigenvectors of matrices for finding community structure in networks [19] and he used modularity in [20]. In [28] authors used a spectral clustering approach for community detection in graph. Their experimental results indicate that the new algorithms are efficient and effective at finding both good clusterings and the appropriate number of clusters. Eigenvectors of network complement reveal the community structure more accurately [29].

Algorithms for finding the community structure in very large networks can be found in [5]. Authors present a hierarchical agglomerative algorithm for detecting community structure. A spectral method for community detection is provided in [16].

3 Spectral Clustering and Ordering

Spectral clustering has become one of the most popular modern clustering algorithms in recent years. It is one of the graph theoretical clustering techniques which is simple to implement. It can be solved efficiently by standard linear algebra methods, and very often outperforms traditional clustering algorithms such as the k-means or single linkage (hierarchical clustering). A comprehensive introduction to the mathematics involved in spectral graph theory is the textbook of Chung [4]; we also recommend the survey of Schaeffer [25]. Spectral clustering algorithm uses eigenvalues and eigenvectors of Laplacian of similarity matrix derived from the data set to find the clusters. A practical implementation of the clustering algorithm is presented in [3]. Recursive spectral clustering algorithm is used in [6]. There Dasgupta et al. analyzed the second eigenvector technique of spectral partitioning on the planted partition random graph model, by constructing a recursive algorithm. A spectral clustering approach to finding communities in graphs was applied in [28].

Ding and He in [7] showed that a linear ordering based on a distance sensitive objective has a continuous solution which is the eigenvector of the Laplacian. Their solution demonstrate close relationship between clustering and ordering. They proposed direct K-way cluster assignment method which transforms the problem to linearization the clustering assignment problem. The linearized assignment algorithm depends crucially on an algorithm for ordering objects based on pairwise similarity metric. The ordering is such that adjacent objects are similar while objects far away along the ordering are dissimilar. They showed that for such an ordering objective function, the inverse index permutation has a continuous (relaxed) solution which is the eigenvector of the Laplacian of the similarity matrix.

3.1 Modularity - Quality of Detected Communities

To quantify the quality of the subdivisions we can use modularity [22], defined as the fraction of the edges in the network that connect vertices of the same type (i.e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices. Networks with the high modularity have dense connections between the nodes within community, but sparse connections between the nodes in the different communities. Modularity is often used in optimization methods for detecting community structure in the networks [20]. The value of the modularity lies in the range $(-0.5, 1)$. It is positive, if the number of edges within groups exceeds the number expected on the basis of chance.

For a *weighted graph* G we have a weight function $w : E \rightarrow R$. It is for example function of the similarity between the nodes v_i and v_j . The weighted adjacency matrix of the graph is the matrix $W = (w_{ij})$ $i, j = 1, \dots, n$. Then the degree of a vertex $v_i \in V$ in weighted graph is defined as

$$d_i = \sum_{j=1}^n w_{ij}.$$

The weighted degree matrix D is defined as the diagonal matrix with the weighted degrees d_1, \dots, d_n on the diagonal.

In terms of the edge weights, modularity $Q(C_1, \dots, C_k)$ is defined over a specific clustering into k known clusters C_1, \dots, C_k as

$$Q(C_1, \dots, C_k) = \sum_{i=1}^k (e_{ii} - \sum_{j=1, i \neq j}^k e_{ij})$$

where $e_{ij} = \sum_{(u,v) \in E, u \in C_i, v \in C_j} w(u, v)$ with each edge $(u, v) \in E$ included at most once in the computation.

4 Left-Right-Oscillate Algorithm for Community Detection

Upon completing our study of various modifications of algorithms for spectral clustering, we designed our own algorithm for detecting communities within complex networks. This algorithm utilizes spectral ordering where similar vertices are closer to indexes and less similar vertices are further from indexes. When determining the ordering, it is necessary to calculate the eigenvector of the second smallest eigenvalue of the matrix $L = D - W$ (D and W have been defined in section 3.1). Since we have designed our algorithm for large amounts of data in a complex network, we used Lanczos method [14] to calculate the approximation of Fiedler vector. Once the Fiedler vector was calculated, we detected appropriate gaps that divide the vertices of a graph into communities. As observed in our experiments, this type of separation into gaps leads to several badly-assigned subgraphs. This is due to the fact that the Fiedler vector is only linear ordered, as is revealed in our data collection. We have designed a Left-right algorithm (see Algorithm 1) for incorporating small subgraphs into larger communities. This approach gradually increases modularity in a given calculation.

Algorithm 1. Left-Right Algorithm for Community Detection

Input: similarity matrix $W = w_{i,j}$ for $i = 1, \dots, n$ and S_c size of smallest communities.

Output: communities C_k , modularity of detected communities

1. We create Laplacian $L = D - W$ using a matrix of similarity W of a connected graph $G = (V, E, W)$.
 2. We calculate the Fiedler vector (the second eigenvector of Laplacian).
 3. We reorder vertices according to Fiedler vector.
 4. We calculate the sums of anti-diagonals $Asum_i = \sum_{\forall j} w_{i-j, i+j}$ a $Asum_{(i \pm 1/2)} = \sum_{\forall j} w_{i-j, i+j \pm 1}$ for all $i = 1, \dots, n$ and we determine $sum_i = Asum(i - 1/2)/4 + Asum(i)/2 + Asum(i + 1/2)/4$.
 5. We approximate the discrete function $Asum_i$ by spline and we determine its first and second derivation. We then find all local minimums and maximums.
 6. We assign maximum gaps that lie between two local maximums. We divide the set of vertices according to its gaps. We obtain subsets $SS_k \subset V$, where $k = 1, \dots, K$ is the amount of subsets.
 7. Using the Left-Right oscillate assigning algorithm (see Algorithm 2), we detect a community.
-

Spectral ordering minimizes the sum of weighted edges multiplied to the power of the difference in index nodes with the edge incidence. The calculation used for this equation is the given eigenvector of the second smallest eigenvalue (Fiedler vector) matrix $L = D - W$. A visualized Fiedler vector and ordered matrix similarity (in agreement with the Fiedler vector) reveals the creation of several natural clusters which is assigned by our algorithm.

For finding the Fiedler vector of Laplacian above a large, sparse and symmetric matrix representative of the evaluated network, we used Lanczos method to partially solve the eigenvalue problem. In our experiments over DBLP for the computation

Algorithm 2. Left-Right-Oscillate Assigned

Input: subsets $SS_k \subset V$, where $k = 1, \dots, K$, S_c size of smallest communities.

Output: communities C_k , modularity of detected partitioning.

1. For subset SS_1 we find connected components C_j , which are greater than the selected size $|C_j| \geq S_c$. These components create communities. We add the rest of the vertices $v_i \in SS_1 - \cup C_j$ to the next subset of vertices SS_2 .
 2. For every subset SS_k $k = 2, \dots, K$ we find next connected components C_j , which are greater than the selected size $|C_j| \geq S_c$. These components create communities. We attempt to assign other vertices to the previous community, which was established in the previous step. If the vertex has no edge leading to the previous community than we add the vertex to the next subset of vertices SS_{k+1} . We continue repeating this method 2 until we reach the end of ordered vertices.
 3. Once we go through all subsets of vertices, connected components are assigned to C_j for $j = 1, \dots, J - 1$ and C_J contain a set of connected components smaller than the selected size.
 4. We employ the same approach going right-left without "oscillation". We begin with $C_{J-1} = C_{J-1} \cup C_J$.
-

of approximated Fiedler vector, the dimension of Krylov subspace was restricted by the size of computer memory.

The next step for the algorithm is to order indexes of vertices $v_i \in V$ for all $i = 1, \dots, n$ in compliance with ordering using Fiedler vector values. Because we want to find communities that are easily detected in a visual representation when ordered by a similarity matrix, we must determine where one community in a linear order ends and the next begins (find two nodes that belong to various communities). For this reason, we have calculated the value of antidiagonal sums above an ordered the set of vertices that capture a cluster overlap in neighboring vertices v_i . We define

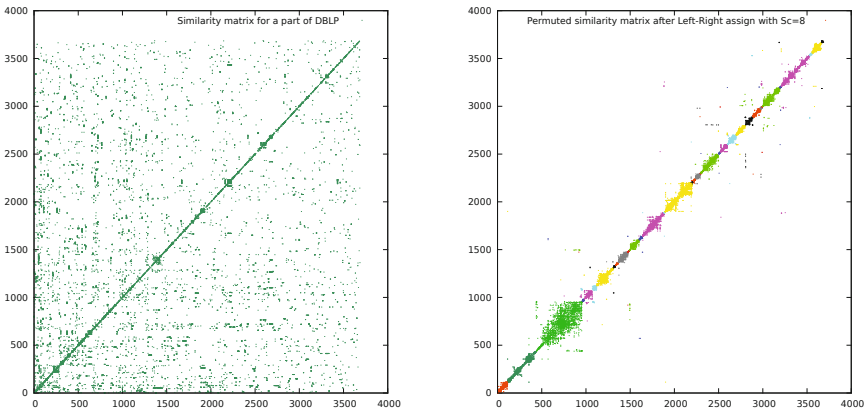


Fig. 1 Similarity Matrix and Permuted Similarity Matrix (Natural Number of Communities is 34)

cluster crossing as the sum of a small fraction of the pairwise similarities. This is aided by linear ordering data points. The goal is to find the nodes that lay in the areas with fewer edges. These vertices lie close to locales with minimum function that are attached by approximation of a cluster overlap discrete function $Asum_i$. We assigned this approximation using the spline function, allowing for easy calculations of both the first and second derivation, which are used to assign local extremes. Between the two local maximum extremes of this function, there lie two vertices. In this area, these vertices represent a maximum gap (the difference in their Fiedler vector value). This gap determines the border between two potential communities.

Once we assign a subset of vertices using gaps, and once we have detected connected components from left to right and vice versa (see Algorithm 2), we always calculate the modularity for the obtained separation of graphs into subgraphs. Our results have revealed that our Left-Right-Oscillate (LRO) algorithm increases modularity. The resulting connected subgraphs then create the structure of communities in the graph.

The usual practice in partitioning of graph is to approximate the general K-way partitioning solution by recursive bi-partitioning, where the graph is broken into two parts based on a partitioning measure at each step. We use this approach with LRO algorithm not for bi-partitioning but for multi-partitioning in one step.

We can use our LRO algorithm in the hierarchical way for the next improvement (see Algorithm 3). This approach is usable for a very large network where the structure is very complex and the one level of community detection is not enough.

Algorithm 3. Hierarchical Left-Right-Oscillate algorithm

Input: similarity matrices $W_k = w_{i,j}$ for all connected components, S_c size of smallest communities, $minM$ minimal modularity.

Output: communities C_k , modularity of detected communities

1. We determine all communities for all connected components via our L-R-O algorithm.
 2. We create new connected components from all communities which are greater then S_c and their modularity is greater the $minM$ so, that we cancel edges between communities and then we continue by step 1.
-

4.1 Community Detection in DBLP

In this section are described the experiments with proposed LRO algorithm for the community detection made for the data collection DBLP, the well known Computer Science Bibliography². The characteristics of the co-authors network DBLP are in the Table 1.

Figure 2 is only a part of the DBLP. The visualization of the whole DBLP is very difficult. But our approach allows a hierarchical point of view to the network. Then, we can select a part of the network with people who are interesting for us.

² <http://www.informatik.uni-trier.de/~ley/db/>

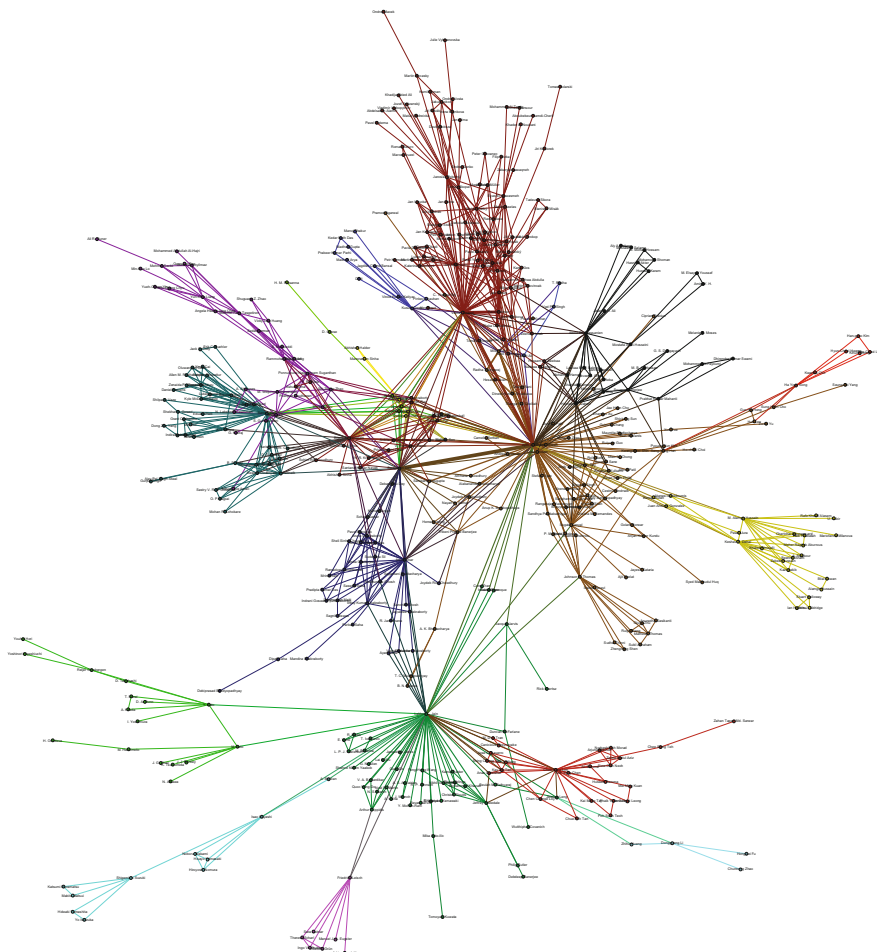


Fig. 2 Selected community with authors (level 8 in the hierarchical LRO algorithm)

Table 1 Characteristics of the DBLP

Number of vertices	1109512
Number of edges	7953382
Number of connected components	100067
Number of isolated vertices	55184
Size of biggest community	919375
Number of articles	1914840
Number of extracted terms	1739421
Average Degree	7.16836
Average Weighted Degree	2.95724

5 Conclusion

In the paper, we have presented the algorithm developed for detection of communities in complex networks. Proposed Left-Right-Oscillate algorithm (LRO) is based on spectral ordering of graph vertices. For more information of our experiments using spectral clustering, see our previous works [23, 26]. In the first phase of this approach, we find the subgraphs using a spectral ordering that is linear. For this purpose, we have used ordering for a given min-cut that we obtain, such as the Fiedler vector of Laplacian ($L = D - W$).

The second algorithm phase then uses the properties of spectral ordering, which re-assigns nodes to indexes based on their proximity (similar nodes to nearby indexes and less similar nodes to further indexes). In the Left-Right phase (Left-Right-Oscillate) we proceed to spectral ordering with gaps, which are separated into subsets of nodes. In this phase, subgraphs with low connectivity whose nodes were separated into subsets of nodes that did not have connectivity, are re-assigned to nearby subsets with connectivity – future communities. This process allows us to identify the natural amount of communities. For large networks such as the DBLP network we can use a hierarchal method. This approach allows us to increase modularity for finding the communities in the original, connected network.

In our future work, we can use spectral ordering for detection of overlapping communities, which more naturally demonstrates the connections between the communities.

Acknowledgements. This work was supported by SGS, VSB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/151 Large graph analysis and processing.

References

1. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An open source software for exploring and manipulating networks (2009)
2. Bruns, A.: How long is a tweet? mapping dynamic conversation networks on twitter using gawk and gephi. Information Communication Society, 1–29 (December 2011)
3. Cheng, D., Kannan, R., Vempala, S., Wang, G.: On a recursive spectral algorithm for clustering from pairwise similarities. Technical report, MIT (2003)
4. Chung, F.R.K.: Spectral Graph Theory, vol. 92. American Mathematical Society (1997)
5. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Physical Review E* 70(6), 1–6 (2004)
6. Dasgupta, A., Hopcroft, J., Kannan, R., Mitra, P.: Spectral Clustering by Recursive Partitioning. In: Azar, Y., Erlebach, T. (eds.) *ESA 2006*. LNCS, vol. 4168, pp. 256–267. Springer, Heidelberg (2006)
7. Ding, C., He, X.: Linearized cluster assignment via spectral ordering. In: *Twenty First International Conference on Machine Learning, ICML 2004*, vol. 21, p. 30 (2004)
8. Dorogovtsev, S.N., Mendes, J.F.F.: *Evolution of Networks: From Biological Nets to the Internet and WWW*, vol. 57. Oxford University Press (2003)
9. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.M.: Self-organization and identification of web communities. *Computer* 35(3), 66–70 (2002)

10. Freeman, L.: Centrality in social networks conceptual clarification. *Social Networks* 1(3), 215–239 (1979)
11. Garton, L., Haythornthwaite, C., Wellman, B.: Studying online social networks. *Journal of Computer-Mediated Communication* 3(1) (1997)
12. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America* 99(12), 7821–7826 (2002)
13. Granovetter, M.S.: The Strength of Weak Ties 78(6), 1360–1380 (1973)
14. Komzsik, L.: *Lanczos Method: Evolution and Application*. Society for Industrial and Applied Mathematics, Philadelphia (2003)
15. Leetaru, K.H.: Culturomics 2.0: Forecasting large-scale human behavior using global news media tone in time and space. *First Monday* 16(9), 2 (2011)
16. Montanari, A.: Community detection via spectral methods. *Signal Processing* (1) (2011)
17. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* 45(2), 58 (2003)
18. Newman, M.E.J.: Detecting community structure in networks. *The European Physical Journal B Condensed Matter* 38(2), 321–330 (2004)
19. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. *Physical Review E - Statistical, Nonlinear and Soft Matter Physics* 74(3 pt. 2), 36104 (2006)
20. Newman, M.E.J.: Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America* 103(23), 8577–8582 (2006)
21. Newman, M.E.J., Barabasi, A.-L., Watts, D.J.: *The structure and dynamics of networks*, vol. 107. Princeton University Press (2006)
22. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear and Soft Matter Physics* 69(2 pt. 2), 16 (2004)
23. Obadi, G., Dráždilová, P., Martinovic, J., Slaninová, K., Snásel, V.: Using spectral clustering for finding students' patterns of behavior in social networks. In: *DATESO*, pp. 118–130 (2010)
24. Pothen, A., Simon, H.D., Liou, K.-P.: Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.* 11(3), 430–452 (1990)
25. Schaeffer, S.: Graph clustering. *Computer Science Review* 1(1), 27–64 (2007)
26. Slaninová, K., Martinovič, J., Novosád, T., Dráždilová, P., Vojáček, L., Snásel, V.: Web site community analysis based on suffix tree and clustering algorithm. In: *Proceedings - 2011 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2011*, pp. 110–113 (2011)
27. Tyler, J., Wilkinson, D., Huberman, B.: E-mail as spectroscopy: Automated discovery of community structure within organizations. *The Information Society* 21(2), 143–153 (2005)
28. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: *Proceedings of the Fifth SIAM International Conference on Data Mining*, vol. 119, p. 274 (2005)
29. Zarei, M., Samani, K.A.: Eigenvectors of network complement reveal community structure more accurately. *Physica A: Statistical Mechanics and its Applications* 388(8), 1721–1730 (2009)

Exploiting Potential of the Professional Social Network Portal “SitIT”

Kamil Matoušek, Jiří Kubalík, Martin Nečaský, and Peter Vojtáš

Abstract. In this paper, we describe the current state of the development of a web portal SitIT.cz. The portal is being developed in the scope of a EU-funded regional project SOSIREČR (<http://www.sosirecr.cz>). It is based on the concept of a social network which has become a very common concept in recent years. It differs from the existing portals in its scope which strictly targets the community of IT specialists in the regions of Czech Republic.

1 Introduction

Our starting point was a fact that the successfulness of each project is based on establishing a good team. Moreover, in the area of the applied research it is absolutely necessary that business cooperates with researchers and academia. However, establishing such kind of cooperation is problematic today, especially in Czech Republic. One of the main reasons of this situation is the lack of communication about the existing offer and demand for IT research and resources (human, finance, etc.). The first concrete goal of the portal is, therefore, to offer an environment for exchanging and sharing information about the existing IT research in the regions of Czech Republic and for meeting the offer and demand for human IT research resources. When doing an IT research, teams need people who perform classical IT professions, e.g., developers or system administrators. Searching for such people is also very problematic for research teams in the Czech Republic. Therefore, the second

Kamil Matoušek · Jiří Kubalík
Department of Cybernetics, CTU Prague, Technická 2, 166 27 Prague 6
e-mail: {matousek, kubalik}@fel.cvut.cz

Martin Nečaský · Peter Vojtáš
Charles University in Prague, Malostranske nam. 25, 118 00 Prague 1
e-mail: {necasky, vojtas}@ksi.mff.cuni.cz

goal of the portal is to offer an environment for meeting the offer and demand for people in classical IT professions.

Of course, it is also necessary to have an overview about the education offered by IT universities and other IT schools in the Czech Republic. It would be valuable to have an overview of the quality of particular study programmes, about their graduates, e.g., how they were satisfied with their programme and where they are employed.

In this paper, we present how the portal achieves the goals mentioned above. The paper is organized as follows. In chapter 2, we describe basic functional requirements in a form of user stories. In chapter 3, we briefly describe the implementation of the portal and the data available in the portal. In the last chapter 4, we discuss extensions to the portal we plan in the future.

2 User Stories

Before we describe the portal itself, we present a few user stories which show the expectations of the users and their requirements. They demonstrate the purposes for which the users can exploit the portal.

2.1 Searching for Research Partners

A hypothetical regional company *ContractsOnline* needs to implement an information system for managing public contracts for cities in its regions. The company found out that there is a lot of different information sources on the Internet offered by the public administration (e.g., business register, information system about public contracts, etc.). It would be very valuable to integrate these sources to the system. The company also learned about an initiative *OpenGov.eu*. The goal of the initiative is to give an open and machine readable access to public administration data to a public.

ContractsOnline has decided that it will scrape the useful data from existing data sources using the techniques mentioned by OpenGov.eu. It studied the web site of the initiative and found out that the main purpose is to represent the published data in a form of the RDF format in the Linked Open Data Cloud¹. Another important aspect is to process existing non-structured or HTML sources provided by the public administration and represent the scraped data in to the form of the RDF format. However, *ContractsOnline* does not have a sufficient know-how in this area. It does not employ experts on RDF and Linked Open Data. Its people do not know the methods of machine processing of unstructured texts. Therefore, it would like to have an access to a portal which would be able to answer the following questions:

¹ <http://linkedopendata.org>

- Which groups or persons in Czech Republic have knowledge about machine processing of unstructured texts, RDF and Linked Data?
- Which groups of persons in Czech Republic cooperate with *OpenGov.eu*?
- Are there any projects in Czech Republic working in the mentioned areas?

There is no sufficient portal on the Internet today. *ContractsOnline* can only use its own network of business contacts or full-text search engines like *Google*. However, the own contact network is too narrow. It does not cover the academia where the required people probably occur. Full-text search results are too large and contain a lot of irrelevant matches. Typically, it is possible to find only a few research teams while more detailed information about their projects is usually hard to trace.

2.2 *Searching for Human Resources*

A hypothetical *department of software engineering (DSE)* was successful in several research project proposals. However, its employees are currently very busy and DSE, therefore, needs to employ new researchers or find some for cooperation. One of the projects requires a J2EEE programmer in the area of mobile applications development. Another project requires an expert on database processing of RDF data. However, the only expert left two months ago. DSE therefore needs a portal which would be able to answer the following questions:

- Who has an experience with research projects in the area of web applications development and has an experience as a J2EE developer?
- Are there any researchers in Czech Republic in the area of database processing of RDF data who publish on relevant conferences?

Similarly to *ContractsOnline*, DSE can use its own network of personal contacts or a full-text search engine. However, none of the options can offer sufficient, actual and complete information about the required persons.

2.3 *Propagation of Research*

A hypothetical XML and web engineering research group (XRG) developed a tool for designing and maintaining a set of XML schemas. The tool is based on several years of theoretical research published at international conferences and journals. The group also developed a set of case studies which demonstrated the usefulness of the tool. Now, it would like to present the tool to a wider network of experts interested in the area and gain a feedback from them. It also searches for a company or companies which could help with transferring the tool to a business practice. The group would use a portal which would offer the following services:

- Publishing the offer of the tool and know-how of the research group. Publishing the offer of the tool and know-how of the research group.
- Dissemination of the offer to potentially interested experts.

The presented user story can also occur in the case of a single researcher who offers his or her expertise to other groups or projects. Similarly to the previous scenarios, a personal contact network is not sufficient. It is also not possible to exploit services offered by various job portals because they do not allow to sufficiently describe the expertise and know-how. Publishing the offer on the web site of the group or a personal website is not very effective because no one is able to read all web sites of all groups and individuals and full-text search engines do not allow to search in such detail. Therefore, publishing the offer is de facto impossible.

3 Portal Implementation

3.1 Basic Entities of the Portal

Let us describe now the entities which can be found at the SitIT portal. Their typical property is that they have their own homepage where they can present themselves, in some cases these pages are further enriched by means of communication and discussion on related topics. In addition to properties with textual information, entities can also be characterized in a structured way by means of several knowledge profiles and scientific profiles (see below).

- *Users* are basic active entities of the portal that are automatically created for all registered users. Users have the option to send messages to other users of the portal and of course also their receiving. In addition, the portal displays *inactive personal entities* which act only in relation to other entities portal, such as members of university management. Their homepages are linked to the respective institutions and they are displayed in a similar way as in the case of active users' homepages, only a limited extent.
- *User groups* are created by active users to support cooperation or just an exchange of ideas on some topic. They can be represented by a group of supporters of modern technology, as well as working groups solving specific research tasks. Those interested in participating in the group must ask the system to add the corresponding link and this link must be confirmed by the group manager. This prevents unwanted users to participate in a private discussion of group members.
- At homepages of *companies*, there can be presented information and published contacts of interesting companies in the field of IT. At the portal launch, there are already established entities of selected companies and other "*active*" *companies* can be created and manage the registered users themselves. Discussion area relevant to these entities is publicly accessible to all users of the portal, e.g. as an opportunity to express opinions on cooperation with the company.
- *Institutions, universities* and other organizations are similar to companies by their data content, but they mostly differ in their orientation, at high schools, there are added links to their organizational units and to information on their study branches. Lists of these organizations are given in advance before the portal

launching and users cannot create new entities. This can be eventually managed on request by portal administrators.

- *Projects* include entities obtained by collecting information about projects supported by public funds in the Czech Republic. In addition, users can create other custom projects they are working on and present the information about them to other users.
- *Advertisements* represent an appropriate space for expressions of supply and demand in the IT field, for example in the case of cooperation on a project. By means of them, users can seek and offer individuals or groups of persons, such as professional teams. At the respective homepages, users can then express their interest; the system informs advertisement managers on it. The portal content is also enriched by current demand with relevant IT jobs by the employment offices of the Czech Republic.
- Presentations of major scientific *journals* at the portal contain useful information mainly on their impact factor, number of articles, etc., and the links to these periodicals.
- In a presentation of *study branch* related to the area of information technology, there are included the corresponding study program together with information about the type of study (e.g., the master’s program), a link to the university, where it is taught, and including the accreditation identifier and the date, until which the accreditation is valid.

Portal SitIT supports certain types of bindings among its entities that can be browsed from the respective homepages of entities. Most of them were described at the preceding list of entities. Generally, for them, the creation or modification of a binding must be confirmed by the other party, i.e. for example a candidate for membership in a particular group is assigned to the group by the portal only after confirmation by the group manager.

User – entity manager has additional rights: he/she can edit the content of entity homepages, acts as a moderator of their discussion areas, the manager can define additional managers of the entities and publish news with eventual attachments in form of files. To observe (i.e. receive) news of selected entities all other users of the portal can subscribe to. Established entities (such as outdated advertisements) can then be removed by him/her.

3.2 Professional Profiles

So called *professional profiles (PPs)* represent a portal specific means for machine-readable description of an extent of knowledge and skills in ICT. Generally, the PPs are structured data types, representing a hierarchy of categories. Currently, we consider two basic types of PPs - a single-layer *knowledge PP* and a tree-structured *scientific PP*. The knowledge PP introduced in [11] consists of 16 plain categories, each of the describing certain ICT knowledge domain, for instance process modelling or data engineering. The scientific PP characterizes an expertise in ICT of

given entity according to the ACM Classification [3] that is universally accepted standard classification of ICT disciplines. The classification has tree structure, each category can be considered as a leaf category or can be further divided into more specific sub-categories. An example of a particular instance of the scientific PP is shown in Figure 1. For both types of PPs, the categories are evaluated using 5-grade scale (0 being the lowest level of expertise/knowledge, 5 being the highest one).

PP can be used in wide range. The scientific PP can be used to characterize user’s expertise, project or research group focus, research teams or projects, etc. The knowledge PP is suitable for describing vacancies offered by firms, a typical profile of a graduate of the study program, etc. Having such sophisticated information management tool, the portal is able to offer functionalities which cannot be offered by existing portals. This includes more exact search results, better characterization of offers and demand of persons in the network, generating analytical outputs (e.g. aggregation of professional profiles by regions, organizations, working positions, etc.), or comparison of professional profiles of selected entities. Most importantly, the concept of the PPs is implemented in a generic way so that any new type of PP can be defined and easily implemented into the portal, please see [2, 4] for details.

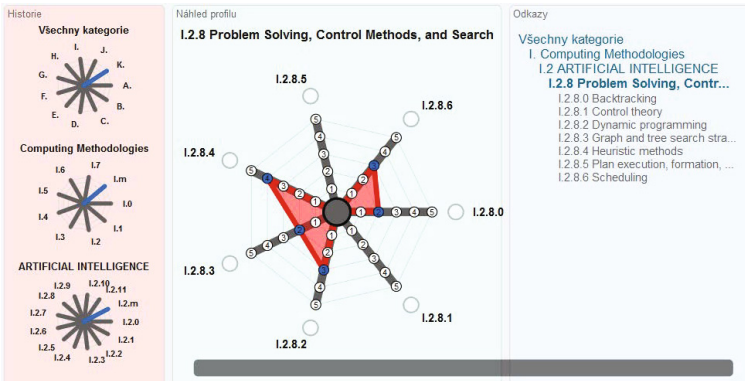


Fig. 1 Example of an instance of the scientific profile. Shown is a particular evaluation within the I.2.8 Problem Solving, Control Methods, and Search ACM node.

3.3 User’s Homepage

General layout of pages on the portal will be illustrated on an example of user’s homepage (in particular, the homepage of the logged in user). Homepages of the other entities on the portal are derived from the user’s homepage layout. The webpage is logically divided into four main parts, see Figure 2:

1. A block of the main information about the entity, see the area outlined in red. It contains its name, photo, link to his/her external homepage, a map indicating his/her home region within the Czech republic, and a concise view of his/her

personal professional profile. Besides the default personal PP, a user can define other scientific and knowledge PPs, for instance a PP of his/her branch of study, PP of his/her Master thesis topic, etc.

2. Detailed information (structured into tabs) about the entity is in the area outlined in blue. In the tab *Details* (Detail profilu), a user can specify his/her contact information such as home or work address, email and telephone number. Then there can be inserted other informations grouped into several sections such as a job position, professional skills, education and keywords. Besides the predefined structure of the detailed information, a user can add any textual information possibly with files attached. Tab *News* (Novinky) plays a role of a board where a user can display news. Each piece of news can be accompanied with files. Tab *Bindings* (Vazby) displays bindings among users and other entities on the portal. Typically, it lists bindings between a user and an institution, where he/she works, bindings to projects on whose the user participates, memberships in research groups and bindings showing relations of type ”colleague” between users.
3. A block of tabs outlined in green contains information about the user’s activities on the portal. Tab *News and events* (Sledovane udalosti) displays list of recent activities that happened on the homepages that the user has chosen to follow. List of these followed homepages is in tab *Followed* (Sleduji). A user can add or remove a homepage from the list of followed homepages by clicking the respective icon at the homepage in question. User can also subscribe for receiving news from followed homepages via RSS. Tab *Managed* (Spravuji) shows a list of homepages for which the user possesses the manager-level rights.

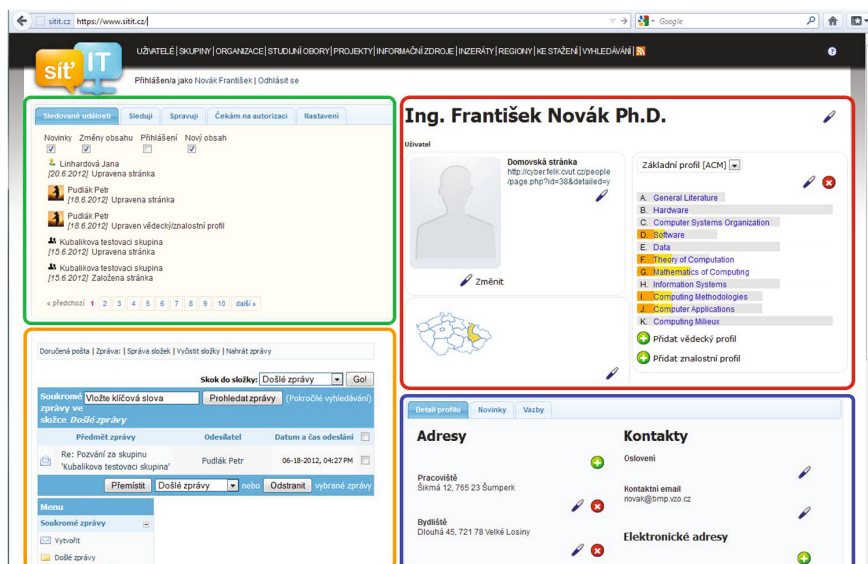


Fig. 2 Homepage of the logged-in user.

4. A component with a simple internal messages management system is in the area outlined in orange.

Note that the tabs in the green frame contain information for the logged-in user, which are permanently displayed to the user irrespectively of the visited homepage that he/she is watching at the current moment. On the other hand, the content of the tabs in the blue frame are always related to the entity presented on the visited homepage. Thus, when a user A is watching his/her own personal homepage then the tabs in green frame will display his/her Details, News etc. Whereas, while watching personal homepage of another user B (or any other entity homepage), the user A would see the tabs in the blue frame with a content relevant to the user B.

3.4 *Trust*

Primarily, the portal should serve as a source of relevant a credible data from the ICT domain in the Czech Republic. Thus, the portal should have some mechanism for dealing with credibility of data. For this purpose, so-called *trust module* for automatic assessing the credibility of users and data inserted to the portal by the users was implemented. Currently, there is one source of trust, called *honesty*, implemented in the portal. It is calculated based on a social proximity of users and explicitly expressed trust in other entities. A variant of the *energy spreading algorithm* [5] was used for calculating the social proximity between two entities. It uses the information gathered from the bindings among users (e.g. two users belong to the same research group, two users are colleagues, etc.). A user can also explicitly express his/her opinion of the homepage of another user or he/she can evaluate professional profiles and news presented by other users. Note, the evaluation is interpreted as agreement or disagreement of the user with the presented information. It cannot be confused with the commonly used "like/dislike" evaluation.

The portal provides users with two possibilities for searching the data - full text searching and search using the professional profiles. In both cases, the results of the search are ranked based on the data relevance to the specified query while taking into consideration the value of the trust assigned to the data by the portal.

3.5 *Communication on the Portal*

A key functionality that systems like this portal have to provide is the means for communication among users. There are several ways of contacting and communication with other users on the portal:

- Each entity except the user has assigned a discussion space. Users can define their own discussion threads within the discussion space of the respective entity. All discussions but the user group one are publicly available for all users. The discussions within the group of users entity are considered private just for members of the respective group.

- For easy reacting on advertisements, users can send a message to the author of an advertisement directly from the homepage of the respective advert.
- Similarly, it is also possible to send a message to another user directly from his/her homepage.

3.6 Data on the Portal

The portal SitIT is pre-filled with content gathered in the framework of SoSIREČR project, which is up-to-date on the day of portal launch for the public. Currently, the portal contains this data: 449 institutions and universities, 65 companies, 241 magazines, 224 projects, 705 study branches, and 23 providers of grant funding.

Its users will surely wonder how updating of this information will be managed. The development team strives to ensure that, where possible, relevant information could be updated automatically via data interfaces used by configurable support modules, or at least to be able to automatically inform the system operator (e.g. via portal or e-mail) in such cases when, for example links on the portal to some web-pages are no longer valid or available. Finally, there is a passive way of informing the system administrators by sending an e-mail to remedy the situation. If we look at individual pieces of content, current status and our ideas are as follows:

- *Users* - these are active entities of the portal, registered users alone are responsible for the timeliness of the data listed on their homepages
- *Groups* - active entities of the portal, group managers are responsible for the timeliness of the information.
- *Projects* - information gathered on projects supported by public funds at the Czech Republic - the possibilities of automatic updates are still analyzed here. In addition, users can create custom entities of other projects. Maintaining their homepages is then the responsibility of those users - managers.
- *Universities and colleges* - we implement an automated support for updating the data within the CTU, potential support for other subjects will be analyzed.
- *Institutions, funders* - too dynamic changes are not expected here, so an update is not carried out so far.
- *Study branches* - update options are analyzed.
- *Companies* - these entities can be established and managed by the registered users themselves, and then these users are also responsible for the timeliness of their entries. Company entities established at system launch can be later passed under the administration of authorized users acting for the company.
- *Journals* - update options especially of more dynamic information such as impact factors and their developments are analyzed.
- *Advertisements* - there are ongoing daily automatic updates of valid open positions of relevant professions from employment offices of the Czech Republic. In addition, users themselves can insert their own advertisements with the types offering individuals or groups and demanding a person or a group. In this case they alone are responsible for the timeliness of data entered.

4 Intended Portal Extensions

- Strengthening regional portal orientation by enabling simple map-based region selection when in the search options.
- Integration of full-text search with searching in knowledge- and scientific profiles in order to support greater variability and more precise queries.
- Enabling to export user's own profile in RDF form according to FOAF (Friend of a Friend) ontology type.

Acknowledgements. This research has been supported by the Czech project GACR P202/10/0761 Web semantization and by the grant no. 7E09078 from the Czech Ministry of Schools, Youth and Sports.

References

1. Vorisek, J., Doucek, P., Novotny, O.: Konkurenceschopnost absolventů IT oborů VŠ a VOŠ na trhu práce v ČR. Hlavní výsledky projektu. Vysoka skola ekonomicka v Praze (May 15, 2007), http://www.vse.cz/media/konkurenceschopnost_it.pdf
2. Kubalik, J., Matousek, K., Dolezal, J., Necasky, M.: Analysis of Portal for Social Network of IT Professionals. *Journal of Systems Integration* 2(1), 21–28 (2011) ISSN 1804-2724
3. The ACM Computing Classification System (1998), <http://www.acm.org/about/class/ccs98.html> (August 4, 2010)
4. Vojtas, P., Pokorny, J., Necasky, M., Skopal, T., Matousek, K., Kubalik, J., Novotny, O., Maryska, M.: SoSIReCR - IT Professional Social Network. In: CASoN 2011, International Conference on Computational Aspects of Social Networks, pp. 108–113 (2011) ISBN: 978-1-4577-1132-9
5. Ziegler, C.N., Lausen, G.: Propagation Models for Trust and Distrust in Social Networks. *Information Systems Frontiers* 7(4-5), 337–358 (2005)

Modeling and Storing Complex Network with *Graph-Tree*

Adan Lucio Pereira and Ana Paula Appel

Abstract. The increased volume of information in recent decades and the emergence of new data types such as complex networks led to the need of development efficient methods for storage and handle these data. Management Systems Database are know for their efficiency and store and retrieve tradicional date as number and small strings. However theses systems need to be modified in order to support complex network data and keep the query processing along with the access methods, the most agile and efficient as possible. Thus the objective of this work is the development of an indexing structure, called *Graph-tree* that can store complex networks to allow binding prediction algorithms to be applied to large complex networks.

1 Introduction

Over the past years the amount of data collected and stored has been substantially increased and it has been powered mainly by the World Wide Web expansion. Part of the data coming from the Web can be represented as graphs, such as page link structures, social and academic networks (Facebook, Orkut, DBLP) and so on. When substantial non-trivial topological features are present in a graph, with patterns of connection between their elements that are neither purely regular nor purely random, the graph-based representation is called a Complex Network [23]. The great amount of data and the new representation approach have motivated the start-up of a research area called graph mining, which has as its main focus investigate, propose and develop new algorithms designed to mine complex networks.

Adan Lucio Pereira
Federal University of Espírito Santo
São Mateus, Rodovia BR 101 Norte, Km. 60, Bairro Litorâneo, CEP 29932-540,
São Mateus - ES - Brazil
e-mail: adanlucio@gmail.com

Ana Paula Appel
IBM Research Brazil, Rua Tutoia, 1157 - Térreo - São Paulo, SP - CEP 04007-900 Brazil
e-mail: apappel@br.ibm.com

The study of complex networks revealed useful properties related to data represented by graphs. Such properties reveal relevant common characteristics of different complex networks. Some interesting and relevant properties are: the power-law degree distributions [2], [1], the diameter shrinkage present in evolving networks [19], the Small World phenomenon [22], among others. These patterns help us to understand not only the interaction among human being and social networks [18] but also the dissemination of information and diseases [6], intrusion detection [13] and so on. A nice graph mining review can be found in [23].

The advance of database management systems (DBMS) has met great emerging challenges from the large volume of complex structured data, such as biological, social networks (Facebook, Orkut), academic networks (DBLP), among others. One of the most important tasks to be performed in these data is the efficient search, for example, what are the possible pairs of nodes that are connected in the near future, it is desirable to retrieve this information quickly in a database composed of a large graph.

Traditionally, indexing structures are used to handle different types of data efficiently, from traditional data, ie, those having relational order between them, even spatial and metric data. These structures are responsible for the efficiency of DBMS in response queries and also they are used in various data mining algorithms as clustering [11, 35].

With the increasing development in complex networks mining area, a computational representation in secondary memory suitable for this type of data becomes necessary. Traditional representations of graphs, such as adjacency and incidence matrices are not useful for such applications, since the computational cost of handling them becomes extremely high. Likewise most traditional database techniques can not be apply directly in complex networks.

The implementation of an indexed structure aims not only to store data securely and efficiently, but also does the minimum use of main memory. If it were possible to use a representation as good as the existing structures to indexing for traditional data also in the analysis of complex networks, it would be possible to speed up the process of knowledge discovery in these type of data, as well as answer queries about to the data field data with simple queries. Unfortunately this is not possible with the current state of the art.

Most methods of knowledge discovery on complex networks assume that the dataset is not too large and relatively simple, which makes most of the algorithms work in main memory, which is not always feasible. Thus, the purpose of this work is to use a secondary memory data structure to store complex network data making possible to use graph mining algorithms in large networks with special focus on prediction algorithms.

This paper is organized as follows: Section 2 presents the main definitions used in this work, Section 3 presents the related work, Section 4 describes the proposed work, Section 5 presents the results and Section 6 concludes the work.

2 Definitions

A graph is a useful way to specifying relationships among a collection of items. A graph consists of a set of objects, called nodes, with certain pairs of these objects connected by links called edges. We say that two nodes are neighbors if they are connected by an edge. A complex network is modeled as a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, on which \mathcal{V} represents the number of nodes/vertex, and \mathcal{E} represents the number of edges/links. The traditional way of represent a graph \mathcal{G} computationally is the adjacency matrix, which is a square matrix $\mathbf{A} = N \times N$, with $N = |\mathcal{V}|$, and $\mathbf{A}_{i,j} = 1$ is $(v_i, v_j) \in \mathcal{E}$ and 0 otherwise.

A graph is undirect if $(v_i, v_j) \in \mathcal{E} \Leftrightarrow (v_j, v_i) \in \mathcal{E}$, that is, the edges are unordered pairs. However, in many settings, we want to express asymmetric relationships, for example, A points to B but not vice versa. For this purpose, we define a directed graph to consist of a set of nodes, as before, together with a set of directed edges; each directed edge is a link from one node to another, with the direction being important. Graphs use here will be consider undirected, if they are not they will be transformed to be.

Graphs are useful because they serve as mathematical models of network structures and they appear in many domains, whenever it is useful to represent how things are either physically or logically linked to one another in a network structure. Social networks, in which nodes are people or groups of people, and edges represent some kind of social interaction [4]; and information networks [12], in which the nodes are information resources such as Web pages or documents, and edges represent logical connections such as hyperlinks, citations [27], or cross-references and so on.

Node degree, also called neighborhood, is defined by the amount of incident edges. Another important concept is the triangles that are triples of fully connected nodes. In many networks it is found that if vertex A is connected to vertex B and vertex B to vertex C, then there is a heightened probability that vertex A will also be connected to vertex C. In the language of social networks, the friend of your friend is likely also to be your friend. In terms of network topology, transitivity means the presence of a high number of triangles in the network [33]. Triangles are an important part of prediction algorithms.

3 Related Work

The purpose of the database area can be described briefly as providing efficient and consistent solutions for storage, maintenance and retrieval large collections of data. Within this aim, an efficient recovery in one of the most important topic and it needs scalable algorithms for large volumes of data, ie algorithms not with greater computational complexity than in the number of linear elements data base.

These goals have traditionally been focused for the management of simple data, such as small numbers and strings, where the search for a given element can always be performed with logarithmic complexity, with the base of the logarithm fairly

large. This leads to a high speed access and, of course, a major reason for the success of the area. In the last years have seen a significant increase in the variety of new data types, with particular emphasis on data from the Web as complex networks. However, the complex networks, which are represented as naturally graphs have been treated as little efficient storage. In this area the majority of studies aim to index a database of small graphs, where the main task is to search for similar graphs or subgraphs [32, 3].

The large volume of available data, the low cost of storage and the stunning success of online social networks and web2.0 applications all lead to graphs of unprecedented size. Typical graph mining algorithms silently assume that the graph fits in the memory of a typical workstation, or at least on a single disk; however there are real graphs that violate these assumptions, spanning multiple Giga-bytes, and heading to Tera- and Peta-bytes of data. Within computer science, there currently exist several options for storing data outside of a traditional relational model.

A promising tool is parallelism, and specifically MapReduce [8] and its open source version, Hadoop. MapReduce is a programming framework [8] for processing huge amounts of unstructured data in a massively parallel way. MapReduce is attractive because it provides a simple model through which users can express relatively sophisticated distributed programs, leading to significant interest in the educational community. Briefly, the programmer needs to provide only two functions, a map and a reduce. The typical framework is as follows: (a) the map stage sequentially passes over the input file and outputs (key, value) pairs; (b) the shuffling stage groups of all values by key, (c) the reduce stage processes the values with the same key and outputs the final result. Hadoop is the open source implementation of MapReduce. Hadoop provides the Distributed File System (HDFS) and PIG, a high level language for data analysis [24]. Based on Hadoop, there are graph mining package for handling graphs with billions of nodes and edges such as PeGaSus [15].

An interesting comparison between parallel DBMS and MapReduce is presented in [26]. As they author say the MapReduce model is so simple and does not provide built-in indexes, which means the programmer must implement any indexes that they may want to speed up access to data inside their application. This is not easily accomplished, as the framework's data fetching mechanisms must also be instrumented to use these indexes when pushing data to running Map instances. Also as the authors describe there are a lot of improvements need for MapReduce architecture be highly adopted.

There are other examples of NoSQL architecture such as, BigTable [7], which is a database system created and used by Google, Dynamo is a key-value storage system used extensively by Amazon [9], Cassandra developed by Facebook [16] and Project Voldemort by LinkedIn [31]. On NoSQL graph database that has attracted a lot of attention is Neo4j, which is open source graph databases for all noncommercial uses. It has been in production for over five years. It is quickly becoming one of the foremost graph database systems. According to the Neo4j¹ website, Neo4j is "an

¹<http://neo4j.org>

embedded, disk-based, fully transactional Java persistence engine that stores data structured in graphs rather than in tables". The developers claim it is exceptionally scalable (several billion nodes on a single machine), has an API that is easy to use, and supports efficient traversals. Neo4j is built using Apaches Lucene 3 for indexing and search. Lucene is a text search engine, written in Java, geared toward high performance. However, most of the algorithms used in this database is base on paths algorithms like Dijkstra, A* and so on [10]. However in graph mining these algorithms are not very usefull since for paths the most common mesuare is geodesic mean or ANF [25]. A comparasion of Neo4j with a relational database is presented in [30] and the authors show that Neo4j is not ready for store graph, not only for the type of queries but also by other properties as multiuser and security.

Another kind of work is graph that management is about RDF (Resource Description Framework) data [17]. RDF data is a collection of statements, called triples, of the form $\langle s, p, o \rangle$, where s is a subject, p is a predicate, and o is an object; each triple states the relation between the subject and the object. A collection of triples can be represented as a directed typed graph, with nodes representing subjects and objects and edges represent- ing predicates, connecting subject nodes to object nodes. There a lot of work in web semantic community for the improvement of RDF data management structured, called triple stores [34, 28]. Most existing triple stores suffer from either a scalability defect or a specialization of their architecture for special-type queries, or both.

4 Proposed Work – *Graph-Tree*

The traditional tactic to speed up data recovery operations is to create data index structures, which organizes the data according to some property of the data (a total order relation in the case of traditional data).

An adjacency matrix is an appropriate representation for a number of cases, especially those which required matrix calculations. However, this is not always matrix representation is beneficial. For example, to retrieve all the neighbours of a node is necessary to traverse the corresponding row in the adjacency matrix looking for non-zeros. This operation is $O(N)$, since N is the line length of the adjacency matrix and that in a complex network can mean a long time. Furthermore, most networks are scattered with the majority of nodes of degree one, which makes the adjacency matrix to be inefficient in the use of memory and if the search neighborhood make it even more inapplicable.

The goal of this project is developping an indexing structure called *Graph-tree* based on tree $B+$ [14] for storing the edge list of a complex network. The edge list representation is convenient and efficient of storage space. Moreover, this type of organization you can store features along the edges and easily find an one. The traditional representation of a edges list, for example in a flat file does not allow the quick search for a node. However in a tree structure that is not a problem.

A $B+$ tree is a variant of the multilevel balanced tree B . The main difference is that all data is written in the leaves of the tree, and they are connected together as a linked list to perform queries easily. The internal nodes contain only keys and tree pointers. The primary value of a $B+$ tree is in storing data for efficient retrieval in a block-oriented storage context in particular, filesystems. This is primarily because unlike binary search trees, $B+$ trees have very high fanout (typically on the order of 100 or more), which reduces the number of I/O operations required to find an element in the tree.

Despite being an efficient structure, a $B+$ tree degrades after several insertions. Moreover, the insertion of one edge at a time in a large network is a time-consuming operation even each operation being $O(\log_b N)$, with b as size of the tree node and N total number of elements. Thus, considering the goal is to store large networks, which already has a large number of edges, efficiently the proposed structure *Graph-tree* works with bulk-loading operation [5]. The bulk-load is a traditional technique for inserting large volume of data efficiently in indexing methods. The focus are static network, ie do not grow over time, therefore, the nodes of the tree have their maximum occupancy instead of keeping the proportion of 50% commonly used trees in B and $B+$. However, even using 50% or more for leaf node occupancy a bulk-load could be used.

The bulk-loading operation inserts the data in an orderly manner and all at once which allows to build *Graph-tree* efficiently. Furthermore, as leaves nodes are inserted completely full there is no need of split the leaf nodes of the tree. Each tree node occupies a disk page of 4 Kbytes, and the number of edges that fit into each page is 512. With these changes the inserts become effective and the tree was optimized for the queries, since the number of nodes decreased and thus the number of disk access was reduced.

4.1 Query Definition

The fact that the majority of networks have a high number of triangles is common knowledge. For instance, in complex networks, especially in social ones, friends of friends are friends themselves. Plenty of research has investigated the behavior of triangles on a network and how they can indicate the existence of larger cliques. Cluster coefficient measures the percentage of a node's neighbors that are neighbors to one another. It measures the degree of "cliqueness" of a graph. Thus one of the operations of interest in such a setting is the estimation of the clustering coefficient and the transitivity ratio, which respectively translates to the number of triangles in the graph, or the number of triangles that a node participates in [29].

Other graph mining task as link prediction also are dependent of triangles. This why a link prediction task is define as: "Given a snapshot of a social network at time t , we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t' " [21]. In the last years a large number of works has been as assigning a connection value, called $score(u; w)$, to pairs of nodes $\langle u, w \rangle$ based on a desired graph G . The scores are ranked in a list in

decreasing order of $score(u;w)$ and then predictions are made according to this list. The most direct implementation of this idea for the link-prediction problem is the common-neighbors predictor, under which we define $score(u;w) := |\Gamma(u) \cap \Gamma(w)|$.

The common-neighbors predictor captures the notion that two strangers who have a common friend may be introduced by that friend. This introduction has the effect of "closing a triangle" in the graph and feels like a common mechanism in real life [33].

As showed triangles are a very important task in complex network mining, thus the first query implement was triangle query. To show the efficiency of *Graph-tree* for each tested complex network was recovery all triangles. The results, reported in Section 5 shows the efficiency of *Graph-tree* insertion operation and the triangle query.

5 Preliminary Results

In this section we present the results achieved in the developed structure. For this experiment 7 datasets were selected from SNAP Website², all of them with high number of nodes and edges. A description of each data set is given bellow, more detail about they can be found in [20].

- **soc-LiveJournal:** LiveJournal online social network;
- **soc-Slashdot0922:** Slashdot social network from February 2009;
- **wiki-Talk:** Wikipedia talk (communication) network;
- **cit-Patents:** Citation network among US Patents;
- **web-Google:** Google programming contest, 2002;
- **amazon0601:** Amazon product co-purchasing network from June 1 2003;
- **roadNet-CA:** Road network of California;

Each graph was preprocessed by removing any self-edges, the direction of the edges and the weights whenever needed. The number of nodes and edges of the networks used after the preprocessing are summarized in Table 1. In detail Table 1 shows respectively: data set name, number of edges (each edge is counted once), number of nodes, Mega-bytes used by *Graph-tree*, average time in seconds to insert all edges in the *Graph-tree*, number of triangles ($\# \Delta$) and time in seconds for count all triangles for each graph using *Graph-tree*. Note that all times are average of three execution and the number of triangles should be divided by 3 since each triangle is counted three times (one for each participant node).

The results of Triangle query is presente in Figure 1 and column $\# \Delta$ and Δ Time in Table 1. As we can see the results show that the time is liner for insertion operation, which is pretty fast, and also for query all triangles in the complex network. This makes *Graph-tree* a suitable struture to store and handle large complex networks. If one wants to store *soc-LiveJournal* network in a adjacency matrix it will

² <http://snap.stanford.edu/>

Table 1 Datasets description and query time

Nome	\mathcal{E}	\mathcal{V}	MB	Insert Time	# Δ	Δ Time
Slashdot0902	504,230	1,807,776	7,8	1	551,724	253
Amazon0601	2,443,408	403,394	38	7	11,959,525	742
RoadNet-CA	2,766,607	1,965,206	43	7	362,028	635
Web-Google	4,322,051	875,713	79	11	40,175,709	1,274
WikiTalk1	10,042,818	2,394,385	72	15	27,610,557	2,857
Cit-Patents	16,518,947	3,774,768	255	25	22,545,069	8,579
Soc-LiveJournal	39,079,299	4,847,571	662	105	857,190,792	20,400

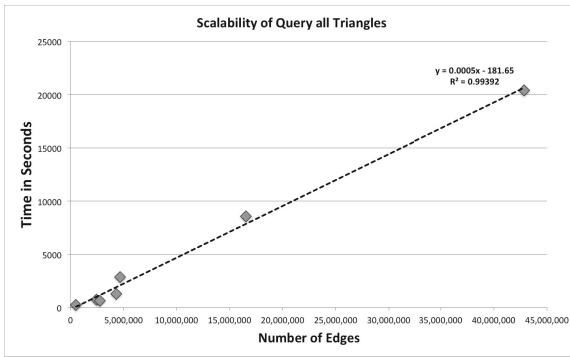


Fig. 1 Retrieve all triangles in a large complex using *Graph-Tree* is linear in the number of edges ($O(E)$)

be necessary 85 Petabytes in memory for store it, which makes this kind of storage unfeasible.

Triangles are a important characterist of complex network and the first step to link prediction algorithm that will be inserted in the struture in the next step as other queries such as page rank, two hops and so on. However for this, it was necessary shows that *Graph-tree* was an efficiency data structure for large graphs that was the propose of this work.

6 Conclusion and Future Work

The main contribution of this work is a new data structure - *Graph-tree* - to store and management large graphs. For now the query allowed in this structure is recovery the triangles of a node or for a network. *Graph-tree* is efficient and run in linear time for insertion and triangle query. Thus, our method can be trivially applied on huge, peta-byte scale complex networks. As future work new queries will be provide such as open triangles, page rank, two hops. A very promising direction is insert *Graph-tree* inside a database management system, such as GIST of postgres.

Acknowledgements. The authors thanks CNPQ and UFES for the support and the reviewers for the useful comments.

References

1. Adamic, L.A., Huberman, B.A., Barab´si, A., Albert, R., Jeong, H., Bianconi, G.: Power-law distribution of the world wide web. *Science* 287(5461), 2115a+ (2000), <http://dx.doi.org/10.1126/science.287.5461.2115a>, doi:10.1126/science.287.5461.2115a
2. Albert, R., Jeong, H., Barabasi, A.L.: The diameter of the world wide web (1999), <http://arxiv.org/abs/cond-mat/9907038>
3. Angles, R., Gutierrez, C.: Survey of graph database models. *ACM Comput. Surv.* 40, 1:1–1:39 (2008), doi:<http://doi.acm.org/10.1145/1322432.1322433>
4. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: King, I., Nejdl, W., Li, H. (eds.) *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011*, Hong Kong, China, February 9–12, pp. 635–644. ACM (2011), doi:<http://doi.acm.org/10.1145/1935826.1935914>
5. De Bercken, J.V., Seeger, B.: An evaluation of generic bulk loading techniques. In: Apers, P.M.G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., Snodgrass, R.T. (eds.) *International Conference on Very Large Databases (VLDB)*, pp. 461–470. Morgan Kaufmann, Roma (2001)
6. Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., Faloutsos, C.: Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur.* 10(4), 1–26 (2008), <http://doi.acm.org/10.1145/1284680.1284681>
7. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.* 26(2), 4:1–4:26 (2008), <http://doi.acm.org/10.1145/1365815.1365816>, doi:10.1145/1365815.1365816
8. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. *Commun. ACM* 51(1), 107–113 (2008), <http://dx.doi.org/10.1145/1327452.1327492>
9. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Voshall, P., Vogels, W.: Dynamo: amazon’s highly available key-value store. *SIGOPS Oper. Syst. Rev.* 41(6), 205–220 (2007), <http://doi.acm.org/10.1145/1323293.1294281>, doi:10.1145/1323293.1294281
10. Dijkstra, E.W.: A Note on Two Problems in Connection with Graphs. *Numerical Mathematics* 1, 269–271 (1959), <http://www-m3.ma.tum.de/twiki/pub/MN0506/WebHome/dijkstra.pdf> (last visited: May 27, 2008)
11. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., Han, J., Fayyad, U.M. (eds.) *Proceedings of the Second International Conference on KDD 1996*, pp. 226–231. AAAI Press (1996)
12. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: *SIGCOMM 1999*, vol. 1, pp. 251–262. ACM Press, Cambridge (1999)
13. Fortunato, S.: Community detection in graphs. *Physics Reports* 486(3–5), 75–174 (2010), <http://dx.doi.org/10.1016/j.physrep.2009.11.002>, doi:10.1016/j.physrep.2009.11.002

14. Johnson, T., Shasha, D.: The performance of current b-tree algorithms. *ACM Transactions on Database Systems (TODS)* 18(1), 51–101 (1993)
15. Kang, U., Tsourakakis, C.E., Appel, A.P., Faloutsos, C., Leskovec, J.: Radius plots for mining tera-byte scale graphs: Algorithms, patterns, and observations. In: *SIAM SDM*, pp. 548–558. Columbus, Ohio (2010)
16. Lakshman, A.: Cassandra - a structured storage system on a p2p network (2012), <http://www.facebook.com>
17. Lassila, O., Swick, R.R., Wide, W., Consortium, W.: Resource description framework (rdf) model and syntax specification (1998)
18. Leskovec, J., Backstrom, L., Kumar, R., Tomkins, A.: Microscopic evolution of social networks. In: *KDD 2008: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 462–470. ACM, New York (2008), doi:<http://doi.acm.org/10.1145/1401890.1401948>
19. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: *Eleventh ACM SIGKDD*, pp. 177–187. ACM Press, New York (2005), doi:<http://doi.acm.org/10.1145/1081870.1081893>
20. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *CoRR abs/0810.1355* (2008)
21. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: *CIKM 2003: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pp. 556–559. ACM, New York (2003), doi:<http://doi.acm.org/10.1145/956863.956972>
22. Milgram, S.: The small world problem. *Psychology Today* 2, 60–67 (1967)
23. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* 45, 167–256 (2003)
24. Olston, C., Reed, B., Srivastava, U., Kumar, R., Tomkins, A.: Pig latin: a not-so-foreign language for data processing. In: *SIGMOD 2008: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1099–1110. ACM, New York (2008), doi:<http://dx.doi.org/10.1145/1376616.1376726>
25. Palmer, C.R., Gibbons, P.B., Faloutsos, C.: Anf: A fast and scalable tool for data mining in massive graphs. In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 1, pp. 81–90. ACM Press, Edmonton (2002)
26. Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S., Stonebraker, M.: A comparison of approaches to large-scale data analysis. In: Çetintemel, U., Zdonik, S.B., Kossmann, D., Tatbul, N. (eds.) *SIGMOD Conference*, pp. 165–178. ACM (2009)
27. Redner, S.: How popular is your paper? an empirical study of the citation distribution (1998), <http://arxiv.org/abs/cond-mat/9804163>
28. Sidirourgos, L., Goncalves, R., Kersten, M., Nes, N., Manegold, S.: Column-store support for rdf data management: not all swans are white. *Proc. VLDB Endow.* 1(2), 1553–1563 (2008), doi:<http://doi.acm.org/10.1145/1454159.1454227>
29. Tsourakakis, C.E.: Fast counting of triangles in large real networks without counting: Algorithms and laws. In: *ICDM 2008*, pp. 608–617. IEEE Computer Society, Washington, DC (2008), doi:<http://dx.doi.org/10.1109/ICDM.2008.72>
30. Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., Wilkins, D.: A comparison of a graph database and a relational database: a data provenance perspective. In: *Proceedings of the 48th Annual Southeast Regional Conference, ACM SE 2010*, pp. 42:1–42:6. ACM, New York (2010), <http://doi.acm.org/10.1145/1900008.1900067>, doi:10.1145/1900008.1900067
31. Voldemort, P.: Project voldemort: A distributed database (2012), <http://project-voldemort.com/>

32. Wang, W., Wang, C., Zhu, Y., Shi, B., Pei, J., Yan, X., Han, J.: Graphminer: a structural pattern-mining system for large disk-based graph databases and its applications. In: SIGMOD 2005: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 879–881. ACM, New York (2005), doi:<http://doi.acm.org/10.1145/1066157.1066273>
33. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. *Nature* 393(6684), 440–442 (1998), doi:<http://dx.doi.org/10.1038/30918>
34. Weiss, C., Karras, P., Bernstein, A.: Hexastore: sextuple indexing for semantic web data management. *Proc. VLDB Endow.* 1(1), 1008–1019 (2008), doi:<http://doi.acm.org/10.1145/1453856.1453965>
35. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: An efficient data clustering method for very large databases. In: Jagadish, H.V., Mumick, I.S. (eds.) ACM SIGMOD International Conference on Management of Data. SIGMOD Record, vol. 25(2), vol. 1, pp. 103–114. ACM Press, Montreal (1996)

Evolution of Author's Profiles Based on Analysis of DBLP Data

Martin Radvanský, Zdeněk Horák, Miloš Kudělka, and Václav Snášel

Abstract. In this paper we introduce a method for analysing the evolution of author's profiles based on keywords extracted from titles of research papers contained in the DBLP database. Academic literature and research papers have been increasing rapidly in recent years. There are a lot of authors who focus each year on popular and widely used topics. Therefore finding experts for particular areas of research is not an easy task. Our solution presented in this paper uses formal concept analysis for finding profiles of author's based on keywords used in titles of research papers. We try to analyse the evolution of these author's profiles over time. Our presented approach is illustrated by using papers contained in the DBLP database from the last decade.

Keywords: DBLP, title keywords, author profile, forgetting function, formal concept analysis, concept stability.

1 Motivation

Digital Bibliography & Library Project (DBLP) is one of the most known collections of electronic resources which can be accessed over the Internet. This project was founded in 1993 and contains, among other things, more than 1,800,000 bibliographic records. These papers come from computer science and were published in different journals and conference proceedings. Although DBLP is primarily used for finding publication in the library, this fast increasing database is often used by

Martin Radvanský · Zdeněk Horák · Miloš Kudělka · Václav Snášel
VSB Technical University Ostrava, Ostrava, Czech Republic
e-mail: martin.radvansky.st@vsb.cz, zdenek.horak.st@vsb.cz,
milos.kudelka,vaclav.snasel@vsb.cz

researchers as a good dataset for data mining. However, the DBLP contains only a limited amount of information about particular papers - there are no abstracts or index terms stored in it. DBLP provides amount of information about the publication activity of authors, conferences and author relationships. There was a lot of research done to find experts, extract their working areas, analyse communities in the social network based on DBLP and much more. We will mention several of them later in this paper in a greater detail.

In this paper we have processed the DBLP in order to extract author's profiles based on keywords used in titles of papers. For our approach we have used Formal Concept Analysis (FCA) and concept stability. We can identify many author's profiles during a one year period so it could be potentially interesting to know how these profiles (w.r.t. to particular keywords) change over the time. We have selected a one-year period and we have tried to find the evolution of selected author's profiles during the selected periods. The expected result of our work is to find author's profiles for the last eleven years using the DBLP database. We would like to consider these profiles as a long-term point of view on the author and his/her research activity.

This paper is organized as follows: Section 2 contains an overview of related work. Section 3 explains the methods used for data evaluation. In Section 4 we describe our data source in a greater detail. Section 5 is focused on finding author's profiles and the evolution of profiles over time. Section 6 concludes the paper.

2 Related Work

Information retrieval and analysis of large social networks is one of the most important research tasks of the last twenty years. A growing number of databases of papers, research papers and other document-oriented databases bring new challenges to the researchers. During the last few years many methods have been introduced focused on the fast searching, grouping and finding similar documents. More recently, the concept of Social Networks appeared. We can look at the database of papers and authors as a social network or even a graph and therefore we can use algorithms from these fields of research.

An efficient algorithm for topic ranking and modelling the evolution of topics was introduced in [11]. The authors show a method for the extraction of keyword sets and cluster the research papers using these keyword sets. Franceschet's paper [2] covers the bibliometrics perspective. It investigates the frequency and impact of conference publications in computer science and compares it with journal papers. The author uses statistical methods for analysing DBLP database. Yan in [12] introduces alternative measures for ranking venues. These measures create new bibliometrics that can be used in ranking publication venues. An application based on

stability (a measure from formal concept analysis which is discussed later) can be found in [8]. In this paper the stability is used for pruning conceptual lattice which was constructed from the European Complex Systems Conference (ECSC) dataset. Analysis of the DBLP database and classification it by using Concept lattices can be found in [1]. This paper shows how concept lattice can cover relational and contextual information of analysed papers. An approach based on forgetting curve and vertex retention and stability in the scientific social network was described in [7].

Our approach is inspired by the previous research, but we have tried to address several issues in a different way. In this paper we try to search the DBLP database for keywords in the titles of papers which are used periodically and frequently by authors. These keywords were clustered by FCA into profiles of authors and finally these profiles were compared over the time. For our experiments we have used the period of the past 11 years. Working with the author's profiles is the main difference of our approach and related work. In the next chapter, we summarize used tools and techniques.

3 Tools and Techniques

This chapter provides some basic notions and techniques applied in our experiments.

3.1 Formal Concept Analysis

In our paper we use Formal concept analysis as a technique for unsupervised clustering. This method helped us to find non-trivial clusters of authors and their keywords.

Formal concept analysis (FCA) is a general data analysis method based on the lattice theory, and it has been introduced in 1982. The basic algorithms for concept lattice computation were published by Ganter in 1984 [3]. More recent publications of these founders can be found in [4, 5, 6].

The input data for FCA we will call formal context C , which can be described as $C = (G, M, I)$ - a triplet consisting of a set of objects G and set of attributes M , with I as relation of G and M . The elements of G are defined as objects and the elements of M as attributes of the context. In the rest of the paper rows of the context correspond to authors and columns correspond to keywords.

As an example of using FCA we have selected five authors a_1, \dots, a_5 and five keywords that were often used by these authors. These keywords are "network" - denoted as k_1 , "analysis" - k_2 , "modelling" - k_3 , "software" - k_4 and "design" - k_5 . The relation between the author and keyword is shown as a cross in the Table 1

Density of the formal context C is defined as proportion of elements of I with respect to the size of $G \times M$. The density calculated for the context depicted in the Table 1 is 52%.

Table 1 Formal context

	network k_1	analysis k_2	modelling k_3	software k_4	design k_5
author a_1	×			×	
author a_2	×		×		×
author a_3		×	×		×
author a_4		×		×	×
author a_5	×		×		

For a set $A \subseteq G$ of objects we define A^\uparrow as the set of attributes common to the objects in A . Correspondingly, for a set $B \subseteq M$ of attributes we define B^\downarrow as the set of objects which have all attributes in B . A formal concept of the context C is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^\uparrow = B$ and $B^\downarrow = A$. The set A is called extent of a concept, while the set B is called intent of a concept. $\mathcal{B}(G, M, I)$ denotes the set of all concepts of context C and forms a complete lattice (so called Galois lattice). For more details see [5, 6]. All concepts from our example are shown in the Table 2. Fig. 1a depicts concept lattice of our example.

For a selection of interesting concepts we have used a method based on concept stability that is described in the next section.

Table 2 Formal concepts extracted from context in Table 1

concept	extent	intent
c(0)	{ a_1, a_2, a_3, a_4, a_5 }	{}
c(1)	{ a_2, a_3, a_4 }	{ k_5 }
c(2)	{ a_1, a_4 }	{ k_4 }
c(3)	{ a_2, a_3, a_5 }	{ k_3 }
c(4)	{ a_2, a_3 }	{ k_3, k_5 }
c(5)	{ a_3, a_4 }	{ k_2, k_5 }
c(6)	{ a_4 }	{ k_2, k_4, k_5 }
c(7)	{ a_3 }	{ k_2, k_3, k_5 }
c(8)	{ a_1, a_2, a_5 }	{ k_1 }
c(9)	{ a_1 }	{ k_1, k_4 }
c(10)	{ a_2, a_5 }	{ k_1, k_3 }
c(11)	{ a_2 }	{ k_1, k_3, k_5 }
c(12)	{}	{ k_1, k_2, k_3, k_4, k_5 }

3.2 Concept Stability

The main problem of using FCA as a clustering method is that we often obtain very large and complicated structure, which is hard to understand and interpret. More technically speaking, we can get a large number of concepts even for a relatively small context. There are several methods which can be used to select only some part of concepts. We have used so-called *concept stability* to filter only the interesting ones. As an interesting concept we consider a concept which is - up to a certain

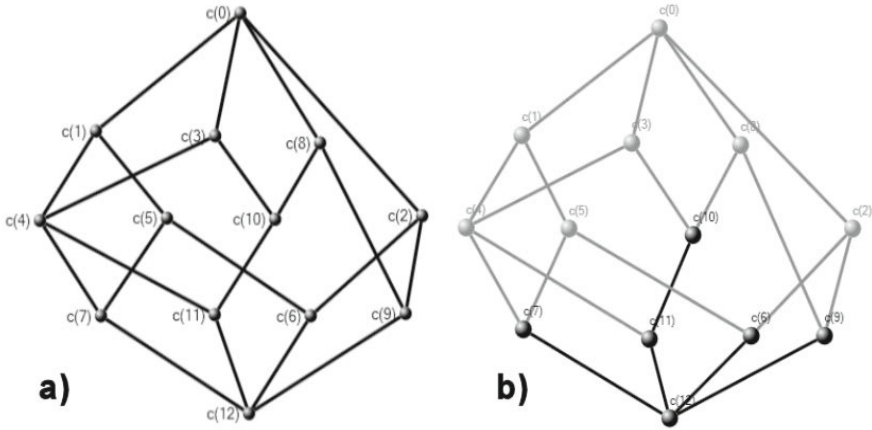


Fig. 1 Concept lattice and pruned lattice created from concepts in the Table 2

degree - resistant to the change of a particular object (removing a particular object does not cause the change of the intent).

Stability of a concept (introduced by Kuznetsov in [8]) expresses the dependency between the intent and extent of the concept. Following the notions from [9], for a particular concept (A, B) of a concept lattice $\mathcal{B}(G, M, I)$, the stability is defined as:

$$\sigma(A, B) = \frac{|\{C \subseteq A | C^\uparrow = B\}|}{2^{|A|}} \tag{3.1}$$

Higher stability causes higher immunity of concept to changes in particular objects. An efficient way to compute the stability of all concepts (using a bottom-up lattice traversal) is described in [10].

To continue our example, we can compute stability of concept $C(10)$ by using Equation (3.1) as:

$$\sigma(\{a_2, a_5\}, \{k_1, k_3\}) = \frac{2}{2^2} = \frac{1}{2}$$

The Fig. 1b shows pruned concept lattice by stability $\sigma \geq \frac{1}{2}$.

4 Data Collection

On December 12, 2011, we downloaded the DBLP database in XML¹ and pre-processed it for further usage. First of all, we selected journal volumes and conferences held by IEEE, ACM and Springer. For every record we identified the month and

¹ Available from <http://dblp.uni-trier.de/xml/>

year of the publication. In the next step we extracted all authors having at least one published paper (997,870 authors). Then we extracted keywords and phrases from paper titles. The extraction approach was based on Faceted DBLP set². In total we used 1,134 keywords and phrases.

We have truncated the selected time period to December 2010 to get the most complete dataset. Then we divided the entire recorded publication period of conferences into one-month time periods. If during one month an author has published a paper, then we set keyword records corresponding to the paper title. For each author we obtained a list of months with occurred keywords. Then we applied the forgetting function (see the remark) to compute the weight of each keyword (or phrases) of each author. For our experiment we have only used keywords and phrases with weight greater than default value (12) at the end of each year processed in our experiments. Next chapter shows evaluation of data collection.

Remark: To calculate the weight (importance) of keywords used in titles of author's papers we have applied forgetting function (detailed description of the function can be found in [7]). This function is based on the simple hypothesis inspired by nature. In our case, we assign a default weight for the word used for the first time. If this keyword is used regularly and frequently, then the weight gradually increases. If not used, the weight is gradually reduced (the word is forgotten). The forgetting allows us to naturally reduce noise in the data.

5 Evaluation of Data Collection

The task of evaluating the collection of data can be described in a few steps.

- From DBLP we have extracted data collection as it was described in chapter 4.
- We used FCA for finding interesting clusters in data collections. We have got formal concepts for each period and we have used pruning of concept lattice by concept stability. All this helped us find author's profiles (see section 5.1).
- The final step of our evaluation was comparing identified author's profiles during selected years (see section 5.2).

In order to create formal contexts from the described data collection we have involved the following transformation: each row of the context represents one author while columns correspond to particular keywords used by authors in the selected period. In the Table 3, the column "Number of authors" shows the number of all authors (w.r.t. the given time-period) which periodically and frequently use keywords in their papers. Column "Selected authors" contains number of authors who have more than one periodically and frequently used keyword.

² <http://dblp.l3s.de/browse.php?browse=mostPopularKeywords>

Table 3 Evolution size of formal contexts

Year	Number of authors	Selected authors	Number of keywords	Context density
2000	1701	122	139	16%
2001	2146	141	157	14%
2002	2962	244	225	10%
2003	3968	370	286	8%
2004	4922	511	352	7%
2005	6945	861	422	6%
2006	8775	1173	491	5%
2007	9456	1218	487	5%
2008	11294	1674	561	5%
2009	11905	1794	561	5%
2010	11355	1735	525	5%

5.1 Finding Profiles of Authors by FCA

Author’s profile by meaning used in this paper is a set of characteristic keywords. These keywords were used often and repeatedly by author in his papers during an observed period. The groups of authors with similar profile, can be seen as a group of experts in the particular research area covered by profile keywords.

FCA gave us a tool for finding profiles of authors based on the keywords they use in the titles of papers. For each context created in the previous steps we have computed a concept lattice. We were interested in non-trivial concepts and author’s profiles are the intents of these concepts. Fig. 2 shows the increasing size of concept lattice.

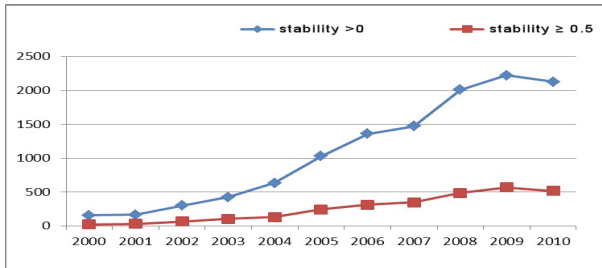


Fig. 2 Evolution of concept lattice size where concepts stability is equal or greater than 0.5

Since not all concepts in the lattice are interesting to us, we have calculated the stability of particular concepts. In the following we have worked with concepts having the stability threshold equal or higher than 0.5 only.

Choosing value of stability threshold has been done according to basic meaning of stability (see section 3.2). Higher value of concept stability makes a concept more confident. It helped us reduce the set of concepts to a size which can be easily explored.

Table 4 Example of author’s profiles that contain keyword database

Year	Authors	Keywords
2000	Sang Hyuk Son, Victor C. S. Lee Rajeev Rastogi	database (0.99), transaction processing (0.75) database (0.99), data mining (0.5)
2005	Paul Watson, Norman W. Paton, Kian-Lee Tan, David Taniar Janet M. Thornton, Sangsoo Kim	database (0.99), query processing (0.82) database (0.99), analysis (0.5)
2010	Victor M. Markowitz, Peer Bork, Yan Zhang, Qing Wang Hans-Peter Kriegel, Sergio Greco, Jeffrey Xu Yu Jiawei Han	database (0.99), analysis (0.78) database (0.99), clustering (0.69) database (0.99), analysis (0.78), clustering (0.69)

Table 4 shows few examples of author’s profiles containing keyword database in some particular years. The numbers in bracket are values of stability.

5.2 Evolution of Authors Profiles

Fig. 3, 4, 5 show particular branches of the previously described concept lattice during several years of the last decade. Each node of the tree represents one concept of the lattice with a focus on concept intents (as formed by publication keywords and their hierarchy). Using these figures we can easily identify the relation between particular keywords and their importance for the whole community (as denoted by the number of authors devoted to keyword combinations). By comparing these figures over different years we can also see the evolution of keywords and their communities through the time.

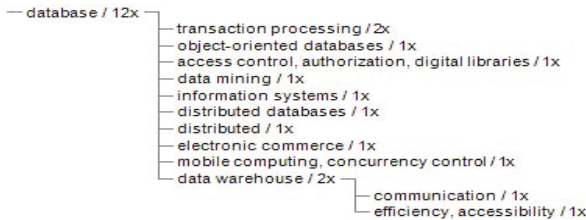


Fig. 3 Hierarchy of papers for keyword “database” during year 2000

As an example we have selected concepts having the keyword “database” in its intent. We can clearly see that the research of the databases has shifted. The beginning of the decade is still devoted to the development of the databases themselves, their paradigms and basic notions, while the figures from the recent years show that the research of the databases is focused more on their applications and interdisciplinary usage. We may consider this situation as "using databases as a tool".

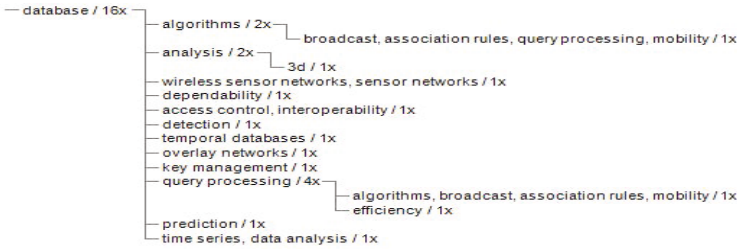


Fig. 4 Hierarchy of papers for keyword “database” during year 2005

The filtering included in the preprocessing phase guarantees that random clusters have been eliminated. Though the size of the relevant data from the beginning of the decade is therefore relatively small, the most important trends are captured and if we focus on the data from last seven years, we get much more complete coverage. We can expect future data to maintain this trend.

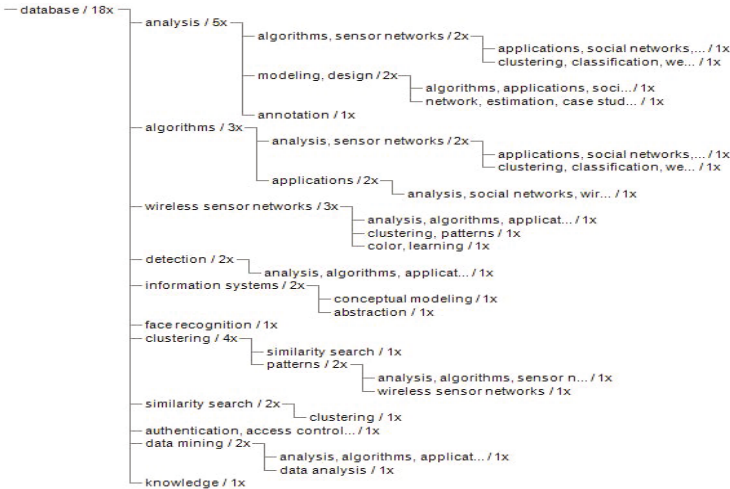


Fig. 5 Hierarchy of papers for keyword “database” during year 2010

6 Conclusion and Future Work

In this paper, we have introduced an approach for finding interesting authors profiles based on keywords used in titles of papers in the DBLP database. We have analysed the evolution of author profiles over the time. For filtering DBLP database we have used so-called forgetting function on the extracted keywords. This approach gives us useful information about the growing author profiles during the last eleven years. Interesting information is the ratio of new authors and authors who published their

papers with the same keywords in the previous years. Another interesting but expected information is the shift of keywords usage, where we can see the trend to use the same keywords as the year before and the decreasing growth of new keywords in the papers which are used periodically and frequently.

Furthermore the FCA method gives us a very interesting hierarchical view on the evolution of keywords connected to the author profiles (presented example illustrates the evolution of one particular field of research over the time). Finally, we have shared our precomputed dataset and exploratory application online (at <http://www.forcoa.net/resources/adbis2012>). In our future work we plan to take a closer look at the cooperation of authors based on their profiles.

Acknowledgements. This paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic, and partially supported by SGS, VSB-Technical University of Ostrava, Czech Republic, under the grants No. SP2012/58.

References



1. Alwahaishi, S., Martinovič, J., Snášel, V., Kudělka, M.: Analysis of the DBLP Publication Classification Using Concept Lattices. In: DATESO 2011, pp. 132–139 (2011)
2. Franceschet, M.: The Role of Conference publications in CS. *Communications of the ACM* 53(12), 129–132 (2010)
3. Ganter, B.: Two Basic Algorithms in Concept Analysis. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 312–340. Springer, Heidelberg (2010)
4. Ganter, B., Stumme, G., Wille, R.: *Formal Concept Analysis, Foundations and Applications*. Springer, Berlin (2005)
5. Ganter, B., Wille, R.: *Applied Lattice Theory: Formal Concept Analysis*. In: Grätzer, G.A. (ed.) *General Lattice Theory*, pp. 592–606. Birkhäuser (1997)
6. Ganter, B., Wille, R.: *Formal Concept Analysis – Mathematical Foundations*. Springer, Berlin (1999)
7. Kudělka, M., Horák, Z., Snášel, V., Abraham, A.: Social Network Reduction Based on Stability. In: CASoN 2010, pp. 509–514 (2010)
8. Kuznetsov, S.O.: On Stability of a Formal Concept. *Annals of Mathematics and Artificial Intelligence* 49(1), 101–115 (2007)
9. Kuznetsov, S.O., Obiedkov, S., Roth, C.: Reducing the Representation Complexity of Lattice-Based Taxonomies. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 241–254. Springer, Heidelberg (2007)
10. Roth, C., Obiedkov, S., Kourie, D.: Towards Concise Representation for Taxonomies of Epistemic Communities. In: Yahia, S.B., Nguifo, E.M., Belohlavek, R. (eds.) CLA 2006. LNCS (LNAI), vol. 4923, pp. 240–255. Springer, Heidelberg (2008)
11. Shubhankar, K., Singh, A.P., Pudi, V.: An Efficient Algorithm for Topic Ranking and Modeling Topic Evolution. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part I. LNCS, vol. 6860, pp. 320–330. Springer, Heidelberg (2011)
12. Yan, S., Lee, D.: Toward Alternative Measures for Ranking Venues: A Case of Database Research Community. In: *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, Vancouver, Canada, pp. 235–244 (June 2007)

Towards Effective Social Network System Implementation

Jaroslav Škrabálek, Petr Kunc, Filip Nguyen, and Tomáš Pitner

Abstract. In this paper we present our latest research in the area of social network system implementation. Both business and technological aspects of social network system development is considered. There are many tools, languages and methods for developing large-size software systems and architectures represented by social network systems. However, no research has been done yet to uncover reason behind the selection and usage of such systems in terms of choosing the right architecture and data storage. We describe effective approach of developing specific parts of social network systems with special attention to data layer (using Hadoop, HBase and Apache Cassandra) which forms basis of any social network system and is highly demanding for performance and scalability.

1 Introduction

The *social network* – a millennium’s first decade phenomenon – has enabled users to connect with people they usually never saw personally and to live virtual life, promoted progressive networking, helped people to find a job or just supported gamification   of regular products and services as a very engaging marketing channel.

Jaroslav Škrabálek · Petr Kunc · Filip Nguyen · Tomáš Pitner
Masaryk University, Faculty of Informatics, Lab Software Architectures and IS
Botanická 68a,
602 00 Brno, Czech Republic
e-mail: [xskrabal, xkunc7, xnguyen, tomp}@fi.muni.cz](mailto:{xskrabal,xkunc7,xnguyen,tomp}@fi.muni.cz),
<http://lasaris.fi.muni.cz>

¹ Gamification is defined as the infusion of game design techniques, game mechanics, and/or game style into anything to solve problems and engage audiences.

Size matters

The social network unlike common information systems represents the supreme discipline of software development. No other information system, application or web service could attract millions of users with immensely progressive potential. This unique opportunity cannot be built without deep study of requirements including users as main decision makers from the very early phase and, of course, without careful selection of functions. For designing a social network system, user-centered approach is particularly crucial. A platform, which social network actually is, takes human perspective into account profits from competitive edge. If customers are satisfied, they are more likely to use new services and recommend the platform to other potential users, which enables the growth of the user basis and provide the foundation for future network. The high number of users indicates high popularity of the platform bringing more people in. Furthermore, if people use the product, they provide a feedback especially, they report errors and require new features. This feedback can lead to significant improvements of the social network and as a result, it can enhance platform as a whole [4].

Architecture is the key

Architecture of the social network system is like a backbone. If it is crooked, the growing potential will never be reached and only a small number of pioneers will use the platform for a limited time and then will leave it. Scalability and robustness as well as universal analysis with advanced level of flexibility will ensure future enhancement, and eventually will help to completely change the initial intention if users will require quite different functionality. This is for example the case of Takeplace [2] - a digital and mobile event management platform helping organizers in all phases of event management process. Thanks to the precise definition of resulting core functions followed by the open-minded and foresighted analysis, the Takeplace becomes a platform supporting both organizers and community of event's attendees capable of managing any kind of event of any size from small seminars, consultancy meetings with 20-40 participants to the large conferences with hundreds of attendees ending with trade shows, fairs and festivals inviting thousands people. Soft part of the development such a social network functional requirements, analysis and design is just the first part within the process of social network system development. We need to know not only *What* to develop, but the question *How* is followed immediately.

Persistence

Selecting the proper back-end technology is crucial. In the starting phases of platform adoption by users, it is very easy to handle the demand by standard tools and software approaches one was used to employ in numerous previous projects. The

turning point is different in every project but that time will certainly come and everybody will learn how restrictive our past decisions may be in the case of social network system. This may even cause the end of the previously well-evolving platform. Therefore, it is indispensable to consider modern persistence tools and frameworks like *Hadoop* from the very beginning since it supports large volume data handling. The non-relational, distributed database HBase and NoSQL database solution Cassandra (described later in the paper) will help developers to keep up with the technological development in time and preserve the direction with respect to contingencies.

Mobile platforms

These needs are also accentuated by tremendous advent of modern mobile platforms. While on the front end the development is simplified thanks to the strict usability approaches required by the companies standing behind iOS, Android or Windows Phone 7, the demand is increasing continuously for cloud supporting back-ends and enabling users to have their data accessible anywhere anytime. Regarding the existing social networks *Facebook*, *Twitter*, *Instagram* or *Pinterest* the platforms the most people speaks about have around 50 % or more traffic just from the mobile smartphones and tablets². Tablets are increasingly appearing and starting to use as the main working tool and this change of ICT utilization paradigm³ will cause in next five years enormous demand for well-developed, not only social network services and platform solutions, on the backend capable to handle millions of inquiries as well as huge data storages.

2 Technological Demands of Social Networks and Case Study

In the domain of social network services data-oriented architectures and technologies are widely used as those services demand high throughput, they are designed for heavy loads, concurrent requests and database could store billions of rows. This section introduces Hadoop and HBase key features and describes them on a real application written in Java using HBase as a persistent storage.

2.1 *Hadoop and HBase*

Use of NoSQL databases means that the data loses relations, developers cannot use structured query language, joins, triggers, procedures etc. Here comes the question

² While Facebook and Twitter varies around the 40-60 % mobile accesses, Instagram is purely mobile social network with 90+ % mobile traffic.

why the system architect would want to choose any of NoSQL databases. The main reason is *scalability*.

The following story describes how growing service based on RDBMS usually develops. Initially, the developers move the system from local environment to the production one with predefined schema, triggers, indexes and in normalized form (3NF or 4NF). As the popularity grows, the number of reads and writes increases. Cache service is used to improve the read time (and the database loses ACID) and to improve write time the components of database server are enhanced. New features are added and database schema has to be changed - either de-normalized or query complexity increases. If the popularity still grows, the servers has to be more powerful (thus expensive) or some functionality must be omitted (triggers, joins, indexes). [5]

This is where software framework Hadoop, developed by Apache, is clearly a better solution as it offers automated and linear scaling, automatic partitioning and parallel computing. Hadoop comprises of two basic parts:

- MapReduce
- HDFS (Hadoop Distributed File System)

MapReduce model was introduced by Google. It consists of two phases which both read and write data in key-value format. The map phase resides in dividing the problem into smaller pieces which are then sent by the master node to be computed on other distributed instances. After solving the current problem, data is sent back to the master node which processes the answers and assembles the solution for the original problem. [6] This model is suitable in situations when the application needs to "write once" and "read many". Traditional relational databases are designed for frequent data writes or updates. The MapReduce model is also designed to run on commodity hardware so it deals with node dropouts: Whenever (in the map phase) the node does not answer in time, the master node just reschedules the problem for other instance. As the reader can see, the only bottleneck is the master node but in the MapReduce model there can exist more master nodes.

HDFS was designed to store huge files across the network. The default block size is 64 MiB which improves the seek time and increase transfer rates for vast data.

Finally *HBase* is non-relational distributed database based on the Hadoop framework. Tables are automatically distributed in the cluster in the form of *regions*. Each region is a data subset defined by first row (included), last row (excluded) and region identifier. When the size of data grows so much that it cannot be stored on one machine they are automatically split and distributed (usually using HDFS) across the nodes.

HBase data model

The basic HBase data model is inspired by Google's BigTable. Tables are in fact four-dimensional sorted map which is persistent. The first dimension is a row - any row has predefined (on the table level) second dimension: column families.

Column families can have variable number of columns which contain the data. The fourth dimension can also be used: the version - each column can remember x older versions of data stored in the column. The HBase data model is designed to store billions of rows and millions of columns. Data has no relations so joining the tables is impossible. Keys and values are arrays of bytes so any data can be persisted. The very simple way how to obtain certain data is to access the value of the map by `table["row"]["cf"]["column"]` (and the version can be specified). As previously said, there cannot be used query language so the only possibility how to obtain more rows is to use scanner which fetch rows of some key interval (the row keys are stored lexically so for example rows `xa` to `c`) to obtain all rows starting on `a` to `c` included.

2.2 Takeplace: Architecture Case Study

An example of architecture will be demonstrated on a working example of social network service designed and implemented for event management platform Takeplace [2], however this subsystem can be used in any service. This goal is achieved by using simple interfaces defining the services. Developer should implement communication layer dealing with remote calls. This communication layer has to provide secure user session identifying current user. The system alone consists of three layered structure connected by interfaces. The service layer provides services for external calls. Data access layer retrieve or store data from/in database and its main goal is to transform business objects into data structures and vice versa. The last layer provides basic CRUD methods and creates a simple framework for any non-relational databases or cache (HBase and *Memcached* are used in this project). There are four services available for external calls: Follow (managing interactions among users), Wall (providing interface related to the users' posts), Discussion (comments connected to certain post) and Like (managing users' favorite posts).

2.2.1 Data Model

Data model of the application comprises of three tables: walls, entities and discussions. Table entities (modeling any user) contains five column families: followers, following and blocked contains user ids in columns. The news column family contains ids of posts on news feed (page which shows posts of people user is following). The last column contains redundant data about numbers of followers, following etc. Table walls stores posts created by users in the system. Column family info stores basic information about the post. In text column family is only one column containing the text of the post and likes column family contains in each column user id of person who like the post. Table discussions is similar to the table walls.

While working with non-relational databases the key aspect of design is to choose row identifications as their choice heavily affects performance. The identifications can describe a relation among data and lexical sorting also defines region on which

this information is stored. For example table walls uses concatenation of user id and time of the post so posts are grouped by users and then sorted by time from newest to oldest. Fetching the wall is fast for each user and there is higher probability that it will be stored on the same region server. There are only weak relations among data. The users (entities) have their posts and they have their comments. These relations are displayed in names of keys and developer is responsible for fetching correct data.

The only problem with performance is loading the news feed. These rows of data will be stored across region servers as the row identifier can vary a lot - as was said the id of any post is assembled from user identification and time - so there would be need to load a post by post to be displayed from each followed user's wall (or the system could store the posts in each profile creating great redundancy). In this case it is suitable to use memory caching tool. While sending a new post to the server it is inserted in the cache in minimized form (also time to live is set) and link to it is put in cached news feed to every interested (following) user. We can obtain news feed in two cache queries. First one fetches the list of posts and the second one (batch query) returns the posts. Memcached is a hash table in random access memory providing fast read/write operations. Once the data gets old or Memcached is full the expired posts are deleted first and after them the least recently used.

This software architecture and data model shows how the social subsystem can be implemented using non-relational databases to allow simple horizontal scaling as the data amount grows, high throughput and which can handle heavy data loads.

3 Maintainable NoSQL Data Model Using Apache Cassandra

This section sets goal of showing simple method for developing Data Layer and Data Access Layer for Social Network written in Java using Cassandra NoSQL database [7]. To access the Cassandra we use Hector API [8] - the most mature API for accessing the Cassandra today. Data layer is a crucial part of any implementation of social network. Such implementation has to meet criteria of

1. maintainability
2. usability
3. transparency
4. compile time checking
5. verification

By (3) it is meant that the layer is understandable and possible to comprehend by new developers. This is for the same reason as (1) and (2). Lastly, the (4) and (5) are something more unique. Java programming language is language for which the extensive unit testing is very typical. It is mainly due to the fact that pioneers of the Test Driven Development (TDD) come from Java background. To achieve (4) one has to use well structured, object oriented, Java API to access his data model. The definition of well structured may be disputed, but average level programmer can

relatively easily - after short period of experience with the API - say whether compile time checking may reveal possible problems in this API. If the API doesn't meet this well structured criterion, the developer has to develop general implementation. We have come to the conclusion what means to be the general wrapper. This can be summarized by *Theorem of DAL API Generality*: "The Data Access Layer API implementation is said to be general when the implementation doesn't have to be changed due to business requirements."

The Generality Theorem is the necessary condition that has to be met for maintainable NoSQL Data Model.

The (5) is well known requirement for any piece of software written today. To achieve it, Test Driven Development is suggested by us. In fact, from empirical experience, the TDD works very well with DAL implementations. There are many reasons for that:

- DAL has well defined inputs and outputs
- Implementation (querying, creating, deleting) is more complicated than to test it
- It is time consuming to test DAL manually. Mainly because of needed setup/teardown of test methods. Also note that manual DAL testing is error prone because tester will give a lot of false positives

Justification of reasons why TDD works well with DAL is unfortunately out of the scope of this paper.

Simply by using Java programming language and Cassandra NoSQL database, the solution gets extra properties for free:

- openness
- multi-platformity
- easily test driven
- robust

Both Java and Cassandra NoSQL database are free of charge and multi-platform (running under JVM). The robustness of Cassandra may be claimed because it was used as Facebook's backing storage for inbox search. Finally, it is possible to embed the Cassandra into an automated tests.

The goal of this section is to explain how to achieve data model in a social network, that will comply to the above criteria while leveraging the technologies. Rest of this section is organized as follows. A small part of data model for social network is given. Test driven approach is applied for this model and data access layer (DAO) classes are defined. DAO layer is a way for a programmer to access the actual data. Rest of the section is devoted to explain aspects of DAO layer and TDD in detail to give detailed insight into the techniques.

3.1 Example

In this section we show how data can be queried and basic objects to access the DAL layer of social network. Almost every social network should contain entity

Person. Assume that Person has two attributes: *name* and *email*. Such a data model is implemented in Java by creating POJO (plain old java object) - object that has no dependencies but Java SDK. Listings for *Person.java* shows possible person implementation.

Person.java

```
public class Person implements Serializable {
    private String id;
    private String name;
    private String email;
    //Getters and setters
    ...
}
```

Another part of the data model is Cassandra Layer. To create entity in NoSQL database ColumnFamilies abstraction is used. It is out of the scope of this paper to introduce this concept. The basic idea is that column families are created from source code. This allows good automation and maintainability. Following listing shows such a usage for person class:

CassandraBootstraper.java

```
public class CassandraBootstraper {
    public void recreateKeyspace(){
        ...
        ColumnFamilyDefinition cfd =
            HFactory.createColumnFamilyDefinition(
                keyspace,
                DBConstants.CF_PEOPLE, ComparatorType.UTF8TYPE);
        c.addColumnFamily(cfd);
    }
}
```

Note, that when working with NoSQL database, we are not creating any column definitions. We just define the column family for holding collection of Person data objects. We do not define the schema for attributes (name or email) in any way. Those are add at runtime of the application. Last important piece of code is DAO - data access object. This object is responsible for CRUD operations (Create Read Update Delete) on the entities. There is one DAO for each Entity hence PersonDAO and MessageDAO. Lets take a look at example of PersonDAO. *PersonDAO.java*

```
@Repository
public class PersonDAO extends DAO {
    ...
    public List<Person> findUsersByName(Set<String> names) {
        Rows<String, String, String> result =
            findRows(DBConstants.CF_USERS,
                names, new StringSerializer());
        List<Person> users = parseUsersFromResult(result);
        return users;
    }
}
```

Example shows read method for getting Person data objects from NoSQL database by their names. The PersonDAO extends DAO object that contains utility methods

like `findRows`. The method `parseUsersFromResult` is private method of `PersonDAO` for parsing the `name` and `email` from the database.

3.2 Data Layer

Modeling of NoSQL database is in the effect very complex task. While basic Entity Relationship modeling techniques are well known and studied for years the NoSQL databases require different approach. Data in NoSQL database are highly de-normalized to gain performance. The data also allow great flexibility of adding new attributes to entities in the database. When building social network it is advisable that no proprietary scripts are introduced.

By the Generality Theorem we can create general implementation of *bootstrapping mechanism*. The mechanism is an idea that all the test data + schema of the database will be created using same programming techniques (Java) as used at runtime. Programmer should use dedicated class (in our case we name this class `Bootstrapper`). This class has following responsibilities:

- connect the Cassandra instance
- create schema in empty Cassandra database
- insert test data

`Bootstrapper` helps better maintainability because information about schema are versioned in this `Bootstrapper` class (using Subversion). `Bootstrapper` also helps testability of code, because unit tests can directly invoke `Bootstrappers` methods.

To implement a DAO Layer to access Cassandra Database it is advisable to use Hector API because it gives a lot of enterprise level features out of the box. The API introduces a lot of clutter. That's why the best approach to implement DAO is to create base class with general implementation for following actions:

- `findRows` (columnFamily, keys, resultSerializer) - finds rows with given keys
- `findAllRows` (columnFamily) - all rows in given family
- `findAllObjectRows` (columnFamily, objectColumnName) - deserializes the object from given column
- `deleteColumn`, `addColumn`, `findObjectColumn`

DAO object

By creating this abstraction the developer can tinker it for specific social network. It should be general enough to allow business changes but as close to the architecture of the social network as possible so that it is highly usable.

Another important technique to be used for Java+Cassandra DAO layer is Aspect Oriented Programming (AOP). We introduce *ErrorHandlingAspect* that is responsible for improving error handling on DAL. The AOP is invaluable in these situations. Reasons stem from the nature of DAL. DAO objects have methods specific for the given entity (Person may have `attachMessage`, which is unique for this entity) and it is important that exception is caught inside of these methods to have bigger picture of the error (parameters, name of the method) and not having to inspect the stack

trace. AOP gives us this flexibility. Only thing needed is to declare *AfterThrowing* aspect on all DAO methods. Apart from error handling, by using AOP we can easily log parameters coming into each DAO method.

Test Driven Development is very advisable technique for DAL implementation for applications using NoSQL Database. Because the data are highly de-normalized and unstructured it is easily possible to introduce regression bugs into the code. In-memory Cassandra instance is possible approach to create such a test suite. On the other hand it may be hard to setup true environment in-memory because the settings/dependencies differ and are not totally transparent with regards to standard Cassandra installation. Therefore developers should take into account possibility of connecting externally running Cassandra instance.

4 Conclusion

In this paper we have presented our current research and development results in area of social network system development. We showed proper usage of existing frameworks, languages and rationale behind their usage. As a developer, project manager or businessman, being aware of existing tools, their proper implementation and foresight of the future development with close collaboration with users will help to achieve success of the platform and its establishment on the market.

References

1. Gamification, <http://gamification.org>
2. Takeplace – an Event Management System, <http://takeplace.eu>
3. <http://www.gartner.com/it/page.jsp?id=1826214>
4. Škrabálek, J., Tokárová, L., Slabý, J., Pitner, T.: Integrated Approach in Management and Design of Modern Web-Based Services. In: Information Systems Development. Springer, New York (2011)
5. White, T.: Hadoop: The Definitive Guide. O'Reilly Media (2009)
6. Lin, J., Dyer, C.: Data-intensive text processing with Mapreduce. Synthesis Lectures on Human Language Technologies 3(1), 1–177 (2010)
7. Lakshma, A., Malik, P.: Cassandra - A Decentralized Structured Storage System
8. Echague, P., McCall, N., et al.: Hector – A high level Java client for Apache Cassandra, <http://hector-client.github.com/hector/build/html/index.html>
9. Beust, C., et al.: TestNG testing framework, <http://testng.org/doc/index.html>

Part VI
Social and Algorithmic Issues in Business
Support

Community Traffic: A Technology for the Next Generation Car Navigation

Przemysław Gawęł, Krzysztof Dembczyński, Wojciech Kotłowski,
Marek Kubiak, Robert Susmaga, Przemysław Wesolek,
Piotr Zielniewicz, and Andrzej Jaszkievicz

Abstract. The paper presents the NaviExpert's Community Traffic (CT) technology, an interactive, community-based car navigation system. Using data collected from its users, CT offers services unattainable to earlier systems. On one hand, the current traffic data are used to recommend the best routes in the navigation phase, during which many potentially unpredictable traffic-delaying and traffic-jamming events, like unexpected roadworks, road accidents, closed roads or diversions, can be taken into account and thereby successfully avoided. On the other hand, a number of distinctive features, like immediate localization of various traffic dangers, are offered. Using exclusively real-life data, provided by NaviExpert, the paper presents two illustrative case studies concerned with experimental evaluation of solutions to computational problems related to the community-based services offered by the system.

Keywords: community traffic, satellite car navigation, reliability analysis, travel time prediction.

1 Introduction

The Community Traffic (CT), a crucial part of the NaviExpert Navigation System, is a technology especially designed to interact with its users. CT, representing the next, more advanced generation of rapidly developing satellite-based car navigation systems, collects an assortment of data concerning the

Przemysław Gawęł · Marek Kubiak
NaviExpert Sp. z o. o., Dobrzyckiego 4, 61-692 Poznań, Poland
e-mail: pgawel@naviexpert.pl

Krzysztof Dembczyński · Wojciech Kotłowski · Robert Susmaga ·
Przemysław Wesolek · Piotr Zielniewicz · Andrzej Jaszkievicz
Institute of Computing Science, Poznań University of Technology,
Piotrowo 2, 60-965 Poznań, Poland

current traffic situation, which are stored, processed and finally used to recommend the best routes during the navigation phase. This means that potentially unpredictable traffic-delaying and traffic-jamming events, resulting from unexpectedly started roadworks, road accidents, closed roads or diversions, can be taken into account and thereby successfully avoided.

In order to operate efficiently, the system processes massive amounts of data which can be generally categorized into implicit data (automatically generated by the mobile application) and explicit data (generated purposefully by the community users). Each kind of data needs specialized procedures. For example, the information generated by the users may be, for various reasons, untrue (e.g. because of being outdated). The analysis in this case involves verifying the reliability of the information sources (i.e. the reliability of those who submitted the information). Its computational challenges are illustrated in the first batch of experiments described in this paper.

At the same time, the bulk of the information received by the system is used for navigational purposes, in particular for finding the fastest routes. This also calls for specialized procedures, in particular for a good travel time prediction model. The model must be fairly stable on the one hand, but flexible enough to react to the dynamically changing traffic situation on the other. Its computational challenges are illustrated in the second batch of experiments described in the paper.

Several other commercial navigation solutions exist with similar purpose. For example, the systems Yanosik (yanosik.pl) and Coyote (www.moncoyote.com) offer services that include collecting user messages and utilizing these messages in danger identification procedures, while TomTom HD Traffic (www.tomtom.com/en_gb/services/live/hd-traffic) and Garmin 3D Traffic Live (www.garmin.com/traffic) offer services that include estimating travel times and utilizing these times in route finding procedures. Another example is the system Waze (www.waze.com), which heavily relies on the community of its users and tries to deal with both of the addressed data processing aspects.

Problems posed and solved in such systems (including the CT system), i.e. verifying the reliability of the information sources and, first of all, predicting the travel times, were described and discussed in numerous papers, including papers on different approaches to assessing data source credibility [3, 4, 6, 8] and papers on different approaches to learning prediction models from floating car data [1, 5, 7, 9, 10, 11].

This paper describes selected services offered by the CT system and provides experimental illustration of the two key aspects. Following this introduction, Section 2 presents the different generations of car navigating systems, describing their intrinsic characteristics, while Sections 3 and 4 introduce two exemplary computational problems related to the community-based services offered by the system. The two sections include also two case studies concerned with experimental evaluation of those problems. The paper is concluded in the final section.

2 Navigation Systems

Early navigation systems essentially lacked the functionality of collecting data from their users and reacting to the dynamically changing traffic situation. In these systems, the route finding was based on information stored within the system, with fairly limited updating capabilities.

CT uses a new car navigation technology, one which relies on bidirectional communication between the system and its users. Being a mobile phone-based application, it allows the users to engage into active interaction with the system. In general, one can distinguish two kinds of data exchange in CT: implicit and explicit.

Car floating data, i.e., time-stamped geographical positions of the GPS devices (and thus the vehicles that carry them), are collected and sent to the system implicitly. These raw positions are converted to passages through road segments (i.e. road units between two adjacent junctions) of the underlying road network, which, under proper assumptions, permits the system to draw more or less accurate conclusions regarding the general fluency of the traffic on the segments. The most immediate deductions regard the actual average speeds of the passages. In result, when searching for fastest routes the system may find it advisable to avoid a particular segment in favour of other segments, which may make the route longer, but ultimately faster.

The remaining difficulty in the fastest route planning is the lack of data on passages through segments. Consider a road segment through which no passages have been observed for some recent time. This may imply that there is no traffic there, so redirecting cars to this segment makes good sense. Unfortunately, observing no passages through a given segment may also imply that (owing to some unpredictable traffic situation, e.g. a serious road accident) the segment had been entirely closed for traffic. In this case, redirecting cars through this segment makes no sense.

To deal with this problem, CT allows its users to generate and submit appropriate messages that inform the system (and thereby its whole user community) about specific traffic situations, like new diversions, various road dangers, speed cameras, etc. The submitted messages can generally be categorized into reporting (or confirming) messages and cancelling messages. For various reasons, the different pieces of information submitted by the users may be untrue (for example because they are no longer up-to-date). This is why the systems attempts to verify the received messages. Verifying such kind of information is, in general, a complex problem. The idea actually utilized here is that of verifying the reliability of the information sources (i.e. of those users who submitted the particular pieces of information).

In addition to route finding, navigating, and gathering user reports, the CT system offers numerous other services, like characterizing and visualizing the current traffic state of selected areas in real time or finding approximate geographical position for mobile phones not equipped with GPS functionality (so-called 'cell ID' identification). Some especially interesting services

arise from cooperation with other communities and involve utilizing recommendations supplied by users of those communities, e.g. recommendations of restaurants, supplied by the users of `gastronauci.pl` or recommendations of natural/architectural monuments, supplied by the users of `wikipedia.pl`. Finally, the system's community can also influence many very system-specific issues, like road categorization or navigational messages.

3 Estimating the Reliability of Submitted Messages

This section illustrates the analysis of warning reports against road dangers, speed cameras, and road checks, submitted to the system by the community users. Unfortunately, such submissions are often quite scattered as far as their location is concerned, because different users move in different directions and, additionally, they generate their messages with various delays. In result, locations of warnings that concern the same event may vary considerably. To be useful, however, these reports should be not only true but also as accurate as possible as far as their locations are concerned. Their analysis is therefore twofold. Firstly, the reports are clustered to discover distinct events and, secondly, their reliability is verified. Below, we illustrate the second phase of the analyses.

3.1 Modified Voting

The simplest idea of computing the reliability of a warning against an event involves computing the ratio of positive reports (i.e. messages that report/confirm the existence of the event) to all reports, the procedure referred to as 'voting'. Let n be the number of all reports in a group of reports and pos the number of positive reports in this group. Then the voting reliability of a warning is equal to $\frac{pos}{n}$.

This voting approach may be slightly modified in order to reduce the reliability of warnings characterized with only few reports: one may notice that when there is only one positive report in a group, then the generated warning would receive reliability of 100%. Therefore, the modified voting reliability is computed as $\frac{pos}{n} \times \frac{n+1}{n+2}$.

3.2 Expectation Maximization

Another idea involves building a specialized probability model for the given data generation scenario. All the variables involved in the scenario are binary:

a report is either positive or negative, a warning either exists or it does not. Thus, a probabilistic model is not difficult to establish [4].

Let n_e and n_u be the number of events and users, respectively. Each user u_i , $i = 1, \dots, n_u$, may send a report concerning an event e_j , $j = 1, \dots, n_e$. Let us further assume that we have a set D of such reports represented by binary variables r_{ij} , stating whether a user u_i confirmed or did not confirm the event e_j . The probability of the observed data can be then expressed by:

$$p(D) = \prod_{(i,j) \in D} (p(u_i) [p(e_j)^{r_{ij}} (1-p(e_j))^{1-r_{ij}}] + (1-p(u_i)) [(1-p(e_j))^{r_{ij}} p(e_j)^{1-r_{ij}}]),$$

where $p(e_j)$ is a probability of a positive event e_j (i.e., the reliability of a warning) and $p(u_i)$ is a probability that a user u_i sends a reliable report (i.e., the reliability of the user). Although these parameters are initially unknown, their values may be estimated using the submitted reports. The problem can be formulated and solved by maximizing the likelihood of observed data, $p(D)$, which is the core of the Expectation Maximization (EM) algorithm [2].

3.3 Experimental Study

The two methods of reliability estimation were compared on a set of user reports generated during a nine-month period of 2007 in the area surrounding the city of Poznań. Only reports related to speed cameras were used; 954 reports were available in this setting.

The modified voting and the EM algorithm both use a reliability threshold to filter unreliable warnings. The values of the threshold were varied from 0 to 1 with 0.1 step.

A reference set of warnings (ground-truth) was available in this experiment, as precise information on the existence of speed cameras in the 43 places mentioned in users' reports was acquired. In 29 cases the speed cameras did exist (reliability equal to 100%), while in 14 cases they did not exist (reliability equal to 0%). One may notice, however, that the reference set is not a properly drawn random sample of potential speed camera positions.

To measure the quality of the approaches we use the number of warnings reported by the methods and the mean square error (*MSE*) of the reliability of the reported warnings with respect to the ground-truth. We only consider warnings that matched the ground truth (this makes it an optimistic estimate, as we do not count warnings that are not related to any of the considered 43 potential places).

The results of the experiment are shown in Table 1. Its contents reveals that the EM algorithm significantly outperforms the voting method for all thresholds. It is also worth noting that, starting from the threshold equal to 0.6, the EM algorithm generates 20 ground-truth warnings with perfect precision: *MSE* series approaches 0. A similar case for the voting algorithm

Table 1 Comparison of the two methods for estimating reliability of warnings: the voting method and the EM algorithm

Threshold	Voting		EM	
	#warnings	MSE	#warnings	MSE
0.0	34	0.120	35	0.094
0.1	34	0.120	34	0.096
0.2	34	0.120	31	0.080
0.3	34	0.120	31	0.080
0.4	32	0.118	22	0.022
0.5	30	0.115	21	0.014
0.6	29	0.109	20	0.000
0.7	24	0.052	20	0.000
0.8	16	0.013	19	0.000
0.9	09	0.006	19	0.000
1.0	00	0.000	17	0.000

starts from 0.8, but then the number of matched warnings starts to fall and its drop in MSE is mainly due to that fall.

4 Estimating the Travel Time

This section illustrates the analysis of data for finding fastest routes, which can be effectively found only when the system has access to accurate estimates of travel times for each road segment. In other words, the goal is to predict the vehicle's travel time between two given points on a road network, which, in order to reduce its computational complexity, is cast to that of estimating the travel time on single road segments.

4.1 The Prediction Model

More formally, we formulate the problem as a prediction of an unknown value of the vehicle travel time y_{st} on a particular road segment $s \in \{1, \dots, S\}$ in a given time point t . The task is then to find a function $f(s, t)$ that estimates the value of y_{st} using a set of training samples $\{(y_i, s_i, t_i)\}_{i=1}^N$. We measure the accuracy of a single prediction $\hat{y}_{st} = f(s, t)$ by a loss function $L(y_{st}, \hat{y}_{st})$, which determines the penalty for predicting \hat{y}_{st} when its true value is y_{st} . A reasonable loss function in this case is the squared error loss:

$$L(y_{st}, \hat{y}_{st}) = (y_{st} - \hat{y}_{st})^2.$$

The whole procedure involves constructing two distinct models, which are finally merged into one, combined model.

The first model, referred to as static, is responsible for predicting overall trends in the traffic. It uses a set of past observations, discovering (potentially existing in the data) repeatable traffic flow patterns (e.g. “at every Sunday morning, on a road segment in the city centre, the traffic is low”). This stability constitutes its strength (the ability to predict for the long-term, e.g. with a horizon of a few days), but also its weakness (the inability to react to dynamically changing traffic situation).

This poor reactivity is the main reason for introducing the second model, referred to as dynamic, which exploits recent observations in real-time. Its goal is to use the most recent of the incoming data to improve the short-term predictions of the static model $f_s(s, t)$. The dynamic model is introduced to account for those changes in the traffic that cannot be explained by exploiting its long-term and periodic behaviour.

The resulting model combines the estimates delivered by the static and dynamic models in the following way:

$$f(s, t) = \frac{\lambda}{r_d(s, t) + \lambda} f_s(s, t) + \frac{r_d(s, t)}{r_d(s, t) + \lambda} f_d(s, t), \quad (1)$$

where $r_d(s, t) \geq 0$ is a reliability of the dynamic model f_d for a given segment s and a given time point t , and $\lambda \geq 0$ is a mixing parameter (tuned experimentally). The reliability defines our trust in the dynamic model. If there are only few or no recent observations, then r_d should be set to a value close to zero or to zero, respectively.

In the following, we use simple static and simple dynamic models to illustrate the capability of the combined model to accurately estimate travel times in the traffic network. Despite their simplicity, these two models, when combined, are powerful enough to be used in practical situations.

4.2 The Static and Dynamic Components

The simplest static model, referred to as the global mean, is based on global averaging:

$$f_s(s, t) = l(s) \times \frac{\sum_{i=1}^N y_i}{\sum_{i=1}^N l(s_i)}, \quad (2)$$

where $l(s)$ is the length of the segment s .

The significant improvement of this model can be obtained by using the segment mean model, which averages the travel times on each road segment separately:

$$f_s(s, t) = \frac{\sum_{s_i=s} y_i}{\sum_{s_i=s} 1}. \quad (3)$$

The particular dynamic model f_d is constructed as a time series for each road segment. Prediction $f_d(s, t)$ for a given segment s and a given time point t is computed using previous observations y_{st_i} , $t_i < t$, from segment s . Training data are then represented for each segment $s \in \{1, \dots, S\}$ in the form $(y_{st_1}, y_{st_2}, \dots, y_{st_{N_s}})$, where N_s is the number of observations for segment s . These observations are aggregated by being averaged over a given time interval T (tuned experimentally):

$$f_d(s, t) = \frac{\sum_{t-t_i < T} y_{st_i}}{\sum_{t-t_i < T} 1}. \quad (4)$$

The reliability parameter $r_d(s, t)$ of this model is set to the number of observations from T , i.e., to $(\sum_{t-t_i < T} 1)$. Thus, we can reformulate the final, combined model (II) to:

$$f(s, t) = \frac{\lambda f_s(s, t) + \sum_{t-t_i < T} y_{st_i}}{\lambda + \sum_{t-t_i < T} 1}, \quad (5)$$

which produces as its output a weighted average over the static model and the most recent observations.

4.3 Experimental Study

In the experiments, we use floating car data that cover the area of Poznań with broad surroundings. The area can be defined as a rectangular envelope with side lengths of above 60 km, centred at 52.3964°N 16.8421°E. In the time domain, the observations span three weeks of 2011: from September 12th till October 2nd, collected between 5:00 a.m. and the midnight (i.e. excluding night hours). The entire data set contains about 3.8 million observations. It should be stressed, however, that the observations are sparse and not evenly distributed in time and space.

We split the data into two parts: the training set and the test set. The training set covers the observations collected during the first two weeks, i.e. from September 12th till September 25th, and is used to construct the static model and to tune the λ parameter. The test set covers observations collected during the last week, i.e. from September 26th till October 2nd, and is used to test the overall performance of the models.

We use in total three methods for travel time estimation: the global mean (GM), the segment mean (SM), and the combination of the segment mean with the dynamic model (CM). We take the observations from the last 5, 15, 30, 60, 120 minutes for building the dynamic model and optimize λ in range $[0.0, 5.0]$ with step 0.5.

The results of the experiment are shown in Table 2. The table reveals that the segment mean improves significantly over the global mean, and the

Table 2 Results of the three models on test set. Mean absolute (MAE) and root mean squared error (RMSE) are reported.

Model	MAE [min]	MAE [%]	RMSE [min]	RMSE [%]
GM	0.1818	100.0	0.5464	100.00
SM	0.1322	72.74	0.4710	86.20
CM, $\lambda=0.0$, $T = 5$ min	0.1322	72.74	0.4710	86.20
CM, $\lambda=2.0$, $T = 15$ min	0.1287	70.80	0.4567	83.58
CM, $\lambda=2.0$, $T = 30$ min	0.1260	69.33	0.4430	81.07
CM, $\lambda=2.5$, $T = 60$ min	0.1247	68.61	0.4357	79.73
CM, $\lambda=4.0$, $T = 120$ min	0.1255	69.05	0.4344	79.50

dynamic model improves further over the segment mean. This is due to the adaptive nature of the dynamic model. Interestingly, the best results are obtained for the time interval T equal to 120 minutes.

5 Conclusions

The paper describes the range of services offered by the NaviExpert's Community Traffic system, a next generation interactive technology that uses various kinds of user-supplied data for finding and recommending best routes during the navigation phase. The development of such systems is directed towards building community networks of their users. Interacting actively with the system, the community can provide data of enormous usability. Their most obvious application is in current route finding, which in result becomes much more reactive to unpredictable traffic-delaying and traffic-jamming events. Another, exclusively community-oriented, application is in shaping the system services, the quality of which may be positively influenced by the community's feedback. Still another application, arising from cooperation with other communities, includes utilizing evaluations of pre-defined objects (e.g. points of interest) supplied by users of those communities.

In two small case studies the papers illustrates an experimental evaluation of two important aspects of the complex data processing carried out by the system: the reliability of information submitted by the community, and the flexibility of the travel time prediction. In each case, two different types of methods were tested: a basically simple, but computationally little demanding method (simple voting in reliability estimation and simple averaging in travel time estimation) and a more advanced, but computationally more demanding method (expectation maximization in reliability estimation and combined model in travel time estimation). In both cases the more advanced methods significantly outperformed the simple ones, achieving results

that make these methods useful enough to be used to practical applications, despite their increased computational demands.

Acknowledgements. This research is as a part of the project UDA-POIG.01.04.00-30-066/11-00 carried out by NaviExpert Sp. z o. o., co-financed by the European Regional Development Fund under the Operational Programme ‘Innovative Economy’.

References

1. Billings, D., Yang, J.: Application of the ARIMA models to urban roadway travel time prediction — a case study. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2006, vol. 3 (2006)
2. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39(1), 1–38 (1997)
3. Hilligoss, B., Rieh, S.Y.: Developing a unifying framework of credibility assessment: Construct, heuristics, and interaction in context. *Information Processing and Management* 44, 1467–1484 (2008)
4. Kubiak, M.: Credibility assessment in an on-line car navigation system by means of the Expectation Maximization algorithm. *Foundations of Computing and Decision Sciences* 32(4), 275–294 (2007)
5. Liu, H., van Lint, H., van Zuylen, H., Zhang, K.: Two distinct ways of using Kalman filters to predict urban arterial travel time. In: Intelligent Transportation Systems Conference, ITSC 2006, pp. 845–850. IEEE (2006)
6. Premaratne, K., Nunez, R., Wickramaratne, T., Murthi, M., Pravia, M., Kuebler, S., Scheutz, M.: Credibility assessment and inference for fusion of hard and soft information. In: Proceedings of AHFE (2012)
7. Rice, J., Van Zwet, E.: A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems* 5(3), 200–207 (2004)
8. Tseng, S., Fogg, B.J.: Credibility and computing technology. *Communications of the ACM* 42(5), 39–44 (1999)
9. Van Lint, J., Hoogendoorn, S., Van Zuylen, H.: Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C* 13(5-6), 347–369 (2005)
10. Wan, K., Kornhauser, A.: Turn-by-turn routing decision based on copula travel time estimation with observable oating-car data. In: Transportation Research Board 89th Annual Meeting, 10-2723 (2010)
11. Zhu, T., Kong, X., Lv, W., Zhang, Y., Du, B.: Travel time prediction for oat car system based on time series. In: 2010 The 12th International Conference on Advanced Communication Technology, ICACT, vol. 2, pp. 1503–1508 (2010)

Situational Requirement Method System: Knowledge Management in Business Support

Deepti Mishra, Secil Aydin, and Alok Mishra

Abstract. Software developers have been successfully tailoring software development methods according to the project situation and more so in small scale software development organizations. There is a need to propagate this knowledge to other developers who may be facing the same project situation so that they can benefit from other people's experiences. In this paper, the use of situational method engineering in the requirement elicitation phase is explored. A new, user-friendly and progressive web-based tool, Situational Requirement Method System (SRMS), for the requirement elicitation phase is developed that can assist in the creation, storage, and extraction of methods related to this phase. These methods are categorized according to some criteria. This categorization also helps in searching for a method which will be most appropriate in a given situation. This approach and tool can also be used for other software development activities.

Keywords: Knowledge Management, Situational Method Engineering, Requirement Elicitation, Method.

1 Introduction

The significance of knowledge management is immense in the software development industry. The software industry is knowledge-intensive and requires extensive management of this knowledge [3]. Software systems are getting increasingly complicated today; the knowledge needed for the implementation is vast and unlikely to be held by any individual software developer [25]. Knowledge management ensures that there is an effective sharing and exploitation of

Deepti Mishra · Secil Aydin · Alok Mishra
Department of Computer Engineering, Atılım University, Ankara, Turkey
e-mail: deepti,alok@atilim.edu.tr, secil.aydin@argela.com.tr

accumulated, collective knowledge [3]. Every company in software development should set software process activities for the processes it adopts. According to each project and the needs of the company, they need different set of activities [20].

Method Engineering (ME) is a discipline to study engineering techniques for constructing, assessing, evaluating, managing methods and to study educational techniques for teaching and training method users [24]. Methods are normally general in nature and they cannot be used directly without adapting them according to the characteristics of the project. Because the engineering situation of each information system development (ISD) project is different, engineering methods need to be adapted, transformed or enhanced to satisfy the specific project situation [17]. In addition to the engineering method tailoring, necessary to fit the project situation, a customization of the engineering method for each engineer participating in the project is also required [17]. This is the concern of situational method engineering, where the term situational method is used to refer to a method tailored to the needs of a particular development setting.

Also, once a method is constructed according to the project situation and later applied successfully, it should be available for the future use in some repository so that others can learn from past experiences. It would be much easier if a tool can be used to store methods in the tool repository and later on methods can be searched from this repository. This tool must also provide the reuse of existing methods to create new methods.

In this paper, the use of situational method engineering in requirement elicitation phase is explored. A web-based tool is developed that can assist in creating methods related with this phase. Methods can be constructed from scratch, by extension or assembly and stored in a web-based tool. These methods are categorized according to criteria we have developed. This categorization also helps in searching a method which will be most appropriate in a given situation. This approach and tool can also be used for other software development activities.

This paper is organized as follows: In the next section, related work is described. In section 3, we have explained the use of situational method engineering in requirement elicitation phase and how the criteria are established. In section 4, Situational Requirement Method System (SRMS) tool is explained. SRMS tool comparison with existing tools is presented in section 5. Finally, the paper concludes in section 6.

2 Related Work

Rolland et al. [23] reported that process prescriptions should be selected according to the actual situation at hand. They experienced that a key discriminant factor in real processes is the product situation. This situation has

a strong bearing in selecting the task best suited to handle it and also the strategy to be adopted in carrying out this task. Rolland et al. [23] also proposed to represent task and strategy alternatives as labeled directed graph called a map and its associated guidelines. Ayed et al. [2] proposed an evolution driven method engineering approach aiming to support the evolution of an existing method (the As-Is method) in order to obtain a new method (the To-Be method) better adapted to a given engineering situation and/or satisfying new methodological requirements. A generic model for situational method engineering has been proposed by Ralyte et al. [21] that supports the integration of different existing SME approaches namely Assembly-based [22], Extension-based [5, 6] Paradigm-based [14]. A method repository containing different reusable method chunks is needed to practice method engineering successfully. But existing methods may not be modular in nature and therefore may not be suitable to be stored in method repository. Henderson-Sellers, Gonzalez-Perez and Ralyte [11] examined “method fragment” and “method chunk”, two main candidates for the atomic element to be used in Situational Method Engineering (SME), in terms of their conceptual integrity and in terms of how they may be used in method construction. Also, parallels are drawn between the two approaches. Harmsen et al. [9] made an analogy between information systems development and method engineering. Gupta and Prakash [8] extended this analogy into three main phases: method requirement engineering, method design and method construction and implementation. They introduced a technical document called method requirement specification (MRS) that describes what a method that meets the MRS has to offer. They developed a representation system for an MRS along with a CAME (Computer Aided Method Engineering) tool called MERU. Arni-Bloch [1] analyzed the requirements for a ME tool and detected the capabilities that are not yet provided by existing tools.

3 Situational Method Engineering for Requirement Elicitation Phase

Two biggest problems in software engineering are the process of efficiently and effectively developing requirements and the tools required for creating truly agile solutions that can change as quickly as your clients mind. IEEE [14] defines requirements engineering mainly as the process of studying user needs to arrive at a definition of system, hardware, or software requirements. Writing good, correct, complete and measurable system and software requirements specifications are a major problem nowadays. Requirement elicitation is the first step in requirement engineering process. Requirements elicitation is defined as the process of identifying needs and bridging the disparities among the involved communities for the purpose of defining and distilling requirements to meet the constraints of these communities [25].

Just a variety of techniques can be used for analyzing and designing software solutions, a variety of techniques can be used to understand user and stakeholder requirements. Every technique has some advantages and disadvantages. None of the technique is perfect in every circumstance. Because of the relative strengths and weaknesses of the available approaches, most projects will normally require a combination of several techniques in order to produce quality results [16]. The most important thing is using the most appropriate technique for a project. Clearly requirement elicitation does not occur in a vacuum and is highly dependent on the specific project, organizational and environmental characteristics [4]. In order to do this, situational requirements engineering should be considered because situational requirements engineering combines requirements with their context. We studied different requirement elicitation techniques and suitability of these techniques in different project situations. With this knowledge, we formed certain criteria that classify different requirement elicitation techniques. These criteria are used during method creation as well as during method selection. These criteria are as follows:

- Experience of the requirement engineer: Experience of the engineer who is responsible for method creation and selection affects the success and reliability of the method. Normally requirements engineers select an elicitation technique by using one of these approaches: select an elicitation technique because it is the only one that they know, select their favorite elicitation technique or select an elicitation technique because they understand intuitively that the technique will be effective in the current project situations [13]. So, if an engineer does not have enough experience, he/she can take wrong decisions. In order to prevent this, experience of the engineer should be considered. For example: if the experience of the engineer is more than 5 years, introspection method can be used for requirement elicitation.
- Experience level of the engineer in similar projects: Experience level of the engineer in similar projects also affects the method because similar project experience provides statistical data to the engineer. By using this statistical data from similar projects, engineer can elicit requirements with the most effective method so similar project experience and collecting statistical data is important for requirement elicitation. It is also important for all phases of the software development. For example: if the similar project experience of the engineer is more than 5 years, using questionnaire technique for requirement elicitation can be useful because the technique is beneficial for similar projects.
- Requirement elicitation period: Requirement elicitation period in software engineering process is important because depending on the time duration, the method to be used should be different. Some of the techniques are effective to use in short time period. For example: if the company have an hour to elicit requirement, role playing can be used for requirement elicitation.

- Experience level of the company's customer: This will also affect the method to be selected for the use. If the customer does not have enough knowledge about software and software environment, requirements cannot be easily elicited from the customer. For example: introspection technique is not effective for less experienced customer if the collected requirements from the engineer are entirely different from the customer's. Prototyping is effective for requirement elicitation if the customer is experienced. It is effective because customer can give valuable feedback to the engineer so the requirements will be more mature and healthy. Workshops are also effective for requirement elicitation when the customer is experienced because the ideas given by the customer will be more valuable.
- Possibility of meeting between development team and customer: If the customer and development team can schedule meetings together, it affects the method to be used in the requirement elicitation. Conducting meetings together is important for taking a common decision in requirement elicitation. Common decision and evaluating lots of ideas is more effective for requirement elicitation because it can lead to mature and stable requirements in the early phase of the development. Brainstorming and workshop techniques are also effective for this type of situations.
- Project budget: Budget is an important factor in any project therefore it also affects the tool and the technique to be used for requirement elicitation. If the project budget is less, role playing can be used for requirement elicitation as it is cheap and less time consuming. Moreover, storyboarding can be used for requirement elicitation because it is inexpensive, easy and also user friendly.
- Project's user interaction level: Depending on level of user interaction with the system, the technique for requirement elicitation should be different. Requirements of user interfaces should be clear early in the project. For example: if the user interfaces in the project are more, use cases technique can be used for requirement elicitation. It is effective for the systems with higher user interaction and also useful for testing.
- Project complexity: Not all of the projects have the same level of complexity. Some systems are complex and therefore it is difficult to elicit requirements in one phase. For these type of systems, different techniques should be used for requirement elicitation. Prototyping technique can be used for complex systems since it is a partial implementation of a system which can help developers, users and customers to better understand system requirements.

Although there exist several studies in the literature which explores SME, but there are not many tools to use. In our work, we designed a new, user friendly and progressive tool for requirement engineering phase. Methods can be created from scratch and stored in the tool repository. Methods are considered as a single, whole unit for reuse purpose instead of method chunks because it is important for the user to see complete method so that he/she can understand it and if require can create a new method by extending, updating

or by assembling two methods. If there is a need to exclude some part of an existing method that we are reusing to create a new method, it can be done by using "update method" functionality.

4 Tool Description

There is a need to use a tool to support creation, storage and extraction of methods. So, a tool, Situational Requirement Method System (SRMS), is developed to demonstrate the proposed solution. SRMS is developed with PHP. MySQL database is used to store the method related information. We preferred to develop a web-based tool because the project members can be in different physical locations still they can easily access it. SRMS has two types of actors: user and admin. Use case diagram for SRMS user is shown in figure 1. SRMS admin can perform following additional functions:

- Add new user
- Set new password for user
- Delete user
- Delete Method
- Delete comment

SRMS is user friendly and easy to use. User first must login to the system. According to the user type; main menu is displayed. Methods that are stored can be viewed by View Method module. It is important to see all methods in the repository before creating a new one. In View Method page, there are criteria which are already explained in Section 3. User can search for the methods based on these criteria to best suit the project situation at hand. All methods satisfying given criteria are displayed. If the desired method is not

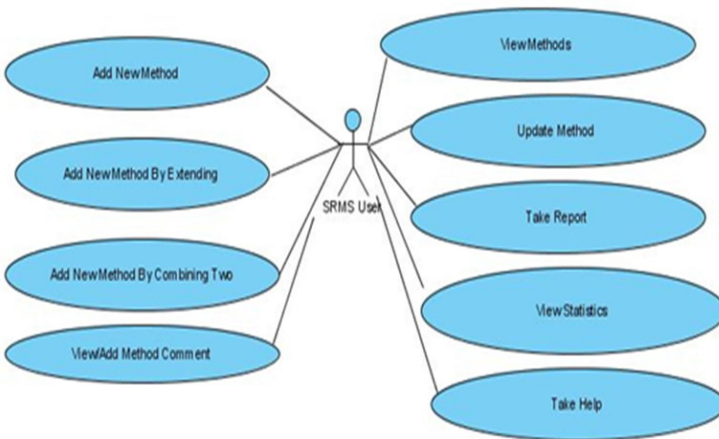


Fig. 1 Use Case Diagram for SRMS User

found in the system, our system enables user to create a new method. There are three different ways for method creation in our system. A method can be created by an empty method template, by modifying a method stored in the repository and by combining two methods stored in the repository. This type of storage provides flexibility to the user to emphasize and transfer his/her experience. If user wants to update an existing method, Update Method module in the main page is used. Only the author of the method can update the method in Situational Requirement Method System.

5 SRMS Tool Comparison with Existing Tools

Although there exist several studies in the literature which explores SME, but there are very few prototypes and tools available in this area. Most of these have been developed for research purpose only. Moreover, majority of these existing tools lack documentation so they are not preferred by the people working in the industry.

CAME tools that exist in the literature are; Decamerone [10], Mentor [26], Meret [12], Meru [8], MetaEdit+ [15], MethodBase [24] and Method Editor [18]. They are compared based on SME approach, method textual language, method meta model, origin, process oriented and product oriented as shown in Table 1.

Decamerone, Meru, MetaEdit+ and Method Editor are assembly-based which involves specifying method requirements, selecting method fragments, and assembling them into a method. Moreover, Meret tool represents a comprehensive methodology representation model. MethodBase aim is to facilitate method customization rather than assembly-based SME. Mentor tool is both assembly-based and paradigm-based. On the other hand, SRMS tool provides method customization from existing methods, by combining two methods according to their context and also constructing a new method from beginning. This approach makes SRMS more flexible for method construction. It also provides more user (experienced engineer) relation with the creation of method so the experience is directly stored in the repository with all characteristics.

Decamerone has its own ontology called Methodology Data Model (MDM) which consists of the basic concepts of information system development products and the associations between them and it has a language for representation of method fragments called method engineering language (MEL). As described before, Meret tool's representation model uses a semantic data-model called ASDM. ASDM provides a powerful means for modeling objects and their interrelationships. Mentor uses the nature contextual approach to describe method fragments. It does not have any language but it has contexts that are pairs of the form <situation, decision>. Meru is based on a meta-model called MVM. In this methodology, method concepts which

Table 1 Comparison between SRMS and existing tools

CAME Envi- ron- ments	SME Ap- proach	Method Textual Lang- uage	Method Meta Model	Origin	Availa- bility	Process Ori- ented	Product Ori- ented	Year of Intro- duction
Deca- meron	Assem- bly- based	MEL	MDM	Research	Web based	✓	-	1995
Mentor	Assem- bly- based, Paradigm- based	-	NATURE	Research	Not web based	-	✓	1996
Meret	Method Cus- tomiza- tion	Method- ology Rep. Model	ASDM	Research	Not web based	✓	-	1992
Meru	Assem- bly- based	MRSL	MVM	Research	Not web based	✓	-	2001
Meta- Edit+	Assem- bly- based	-	GOPPRR	Research and Com- mercial	Cross- platform	✓	-	1994
Method- Base	Method Cus- tomiza- tion	Object Z	-	Research	-	✓	-	1992
Met- hod Editor	Assem- bly- based	MEL	UML	Research	Not web based	✓	-	2003
SRMS	Method Cus- tomiza- tion	English	Complete Method, Criterion used	Research	Web based, cross- platform	✓	-	2009

are called things are partitioned to links, constraints and product elements. Meru has its own language called method requirements specification language (MRSL). MetaEdit+ uses the GOPPRR conceptual data model as its method specification language. GOPPRR composed of graph, object, port, property, relationship and role. MethodBase's data model is divided into product and process parts. It uses Object Z to represent method fragments. Method Editor uses UML for meta-modeling technique to represent method fragments. Class diagrams are used for the specification of product fragments, while process fragments are described by means of activity diagrams. SRMS uses various criteria for storing and representing methods. There is no need to learn a special language or a modeling approach for representing a methods. SRMS stores methods in natural language which provides ease of use as well as flexibility to the users.

Based on origin, Decamerone, Mentor, Meret, Meru, MetaEdit+, Method Editor and SRMS are developed for research purposes. Commercial version

of MetaEdit+ has been released. Mentor, Meret, Meru and Method Editor are not web based tools whereas Decamerone and SRMS are web based tools. SRMS and MetaEdit+ are cross-platform tools. Since SRMS is a web based tool therefore even if project members are in different physical locations, they can easily access it.

Depending on the way CAME environments facilitate the enactment of the method engineering process; they are grouped as product-oriented and process-oriented. The tools that are providing less support for method's process model and its enactment and focuses on product related issues are classified as product-oriented. Process-oriented CAME environments focuses on the process related issues and supports the enactment of the process model. Decamerone, Meret, Meru, MetaEdit+, MethodBase, Method Editor and SRMS are basically process oriented. On the other hand, Mentor is the only product-oriented tool.

Odell [19] and Harmsen et. al. [9] suggest CAME tool should support the seven features; definition and evaluation of contingency rules and factors, storage of method fragments, retrieval and composition of method fragments, validation and verification of the generated methodology, adaptation of the generated methodology, integration with a meta-CASE tool and interface with a methodbase [19, 9]. Tools that are described above satisfy partially these seven features.

6 Conclusion

It has been found that software developers are successfully tailoring standard software development methods according to their project situations. The main problem is that this knowledge and experience is not communicated to other people who might be facing the same project situations especially in small software development organizations. A simple and formal approach to method tailoring, and storage can solve this problem and developers can learn from each other's experiences. This will also enhance their productivity. We have developed a simple approach for method storage and retrieval according to project characteristics with the help of a web-based tool. New methods can be developed from scratch, by extension or assembling existing methods. We applied this approach and tool specifically for requirement elicitation methods but this approach and tool can be applied for any software development phase.

References

1. Arni-Bloch, N.: Towards a CAME Tools for Situational Method Engineering. In: Interop-ESA 2005, Geneva (2005)
2. Ayed, M.B., Ralyté, J., Rolland, C.: Constructing the Lye method with a method engineering approach. *Knowl.-Based Syst.* 17(5-6), 239–248 (2004)

3. Chandani, A., Neeraja, B., Sreedevi: Knowledge Management: An overview & its impact on Software Industry. In: International Conference on Information and Communication Technology in Electrical Sciences, ICTES 2007, pp. 1063–1068 (2007)
4. Christel, M.G., Kang, K.C.: Issues in requirements elicitation, Software Engineering Institute, Carnegie Mellon University, Pittsburg, PA, Technical Report. CMU/SEI-92-TR-012 (1992)
5. Deneckere, R.: Approche d'extension de methods fondée sur l'utilisation de com-posants generiques. PhD thesis, University of Paris 1-Sorbonne (2001)
6. Deneckere, R., Souveyet, C.: Patterns for extending an OO model with Temporal Features. In: Proceedings of OOIS 1998 Conference. Springer, Paris (1998)
7. Dominguez, E., Zapata, M.A.: Noesis: Towards a situational method engineering technique. *Inf. Syst.* 32(2), 181–222 (2007)
8. Gupta, D., Prakash, N.: Engineering Methods from Method Requirements Specifications. *Requir. Eng.* 6(3), 133–160 (2001)
9. Harmsen, F., Brinkkemper, S., Oei, H.: Situational Method Engineering for Information System Project Approaches. In: Verrijn-Stuart, A.A., Olle, T.W. (eds.) *Methods and Associated Tools for the Information System Life Cycle*. Elsevier Science (1994)
10. Harmsen, F.: *Situational Method Engineering*. Moret Ernst and Young Management Consultants (1997)
11. Henderson-Sellers, B., Gonzalez-Perez, C., Ralyté, J.: Comparison of Method Chunks and Method Fragments for Situational Method Engineering. In: Proceedings of the 19th Australian Conference on Software Engineering, ASWEC 2008, March 26-28, pp. 479–488. IEEE Computer Society, Washington, DC (2008)
12. Heym, M., Osterle, H.: A semantic data model for methodology engineering. In: Proceedings of Fifth International Workshop on Computer-Aided Software Engineering. IEEE (1992)
13. Hickey, A.M., Davis, A.M., Kaiser, D.: Requirement Elicitation Techniques Analyzing the Gap Between Technology Availability and Technology Use, Comparative Technology Transfer and Society (2003)
14. IEEE Standard Glossary of Software Engineering Terminology, IEEE (1990)
15. Kelly, S., Lyytinen, K., Rossi, M.: MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment. In: Constantopoulos, P., Vassiliou, Y., Mylopoulos, J. (eds.) *CAiSE 1996*. LNCS, vol. 1080, pp. 1–21. Springer, Heidelberg (1996)
16. Maiden, R.A.M., Rugg, G.: ACRE: selecting methods for requirements acquisition. *Software Engineering Journal* 11(3), 183–192 (1996)
17. Mirbel, I., Ralyte, J.: Situational method engineering: combining assembly-based and roadmap-driven approaches. *Requir. Eng.* 11(1), 58–78 (2005)
18. Niknafs, A., Ramsin, R.: Computer-Aided Method Engineering: An Analysis of Existing Environments. In: Bellahsene, Z., Léonard, M. (eds.) *CAiSE 2008*. LNCS, vol. 5074, pp. 525–540. Springer, Heidelberg (2008)
19. Odell, J.J.: *Introduction to Method Engineering*. Object Magazine (1995)
20. Pressman, R.S.: *Software Engineering: A Practitioner's Approach*. McGraw-Hill (2005)
21. Ralyte, J., Deneckere, R., Rolland, C.: Towards a Generic Model for Situational Method Engineering. In: Eder, J., Missikoff, M. (eds.) *CAiSE 2003*. LNCS, vol. 2681, pp. 95–110. Springer, Heidelberg (2003)
22. Ralyté, J., Rolland, C.: An Assembly Process Model for Method Engineering. In: Dittrich, K.R., Geppert, A., Norrie, M. (eds.) *CAiSE 2001*. LNCS, vol. 2068, pp. 267–283. Springer, Heidelberg (2001)

23. Rolland, C., Prakash, N., Benjamin, A.: A multi-model view of process modeling. *Requirements Engineering* 4(4), 169–187 (1999)
24. Saeki, M.: CAME: The first step to automated method Engineering. In: *Workshop on Process Engineering for Object-Oriented and Component-based Development*, Anaheim, CA (2003)
25. (SEI)-Software Engineering Institute Requirements Engineering Project, *Requirements Engineering and Analysis Workshop Proceedings*, Technical Report, Carnegie Mellon University (1991)
26. Si-Said, S., Rolland, C., Grosz, G.: MENTOR: A Computer Aided Requirements Engineering Environment. In: Constantopoulos, P., Vassiliou, Y., Mylopoulos, J. (eds.) *CAiSE 1996*. LNCS, vol. 1080, pp. 22–43. Springer, Heidelberg (1996)

Effectiveness Analysis of Promotional Features Used in Internet Auctions: Empirical Study

Adam Wojciechowski and Paweł Warczynski

Abstract. Internet auctions sites are trading platform with huge number of visitors that may become customers. On-line auctions allow users to buy and sell products without geographical borders, because in majority of cases ordered products are deliver by surface mail. The aim of our research was to collect substantial amount of data which describe on-line auctions in variety of attributes, as well as customers behaviour. We provide a quantitative arguments in discussion on effectiveness of promotional features used in on-line auctions.

Keywords: On-line auctions, price effectiveness, good practices.

1 Introduction

Internet auctions are subject of extensive study and scientific research in many aspects[5]. They are an interesting marketplace for new and second-hand products but also they may play role of trend makers or provide a reference for current market price [4] on variety of products. However for many beginning and experienced merchants on-line auction sites are often primary marketplace – a trading platform with huge number of visitors that may become customers. On-line auction sites allow users to buy and sell products without geographical borders, because in majority of cases ordered products are deliver by surface mail.

Auction sites, depending on their position on market may offer free or paid services. While basic functionality of an on-line auction is not expensive (or even free) for a seller, additional promotional paid features applied to auctions potentially increasing transaction probability are products where auction site operators try to generate their income. Those advertising features that may bring mutual benefits – for auction sites operators they bring direct profit and for sellers the features, when wisely used, may turn into increase of sales. Although there are plenty

Adam Wojciechowski · Paweł Warczynski
Poznan University of Technology, Institute of Computing Science
ul. Piotrowo 2, 60-965 Poznan, Poland
e-mail: Adam.Wojciechowski@put.poznan.pl, PawelWar@gmail.com

of good practice guides for on-line auction traders they rarely are provided with experimental, numerical data giving clear evidence and statistical confirmation of recommended practices. Our aim is to collect observations draw some statistical conclusions and provide numerical data for individual interpretation or comparison with other on-line auctions or similar experiments conducted in different periods.

2 The Aim of Experiment and Data Collection

The aim of our project [6] was to collect substantial amount of data which describe on-line auctions in variety of attributes, as well as customers behaviour denoted by a collection of performed actions: visits on auction web page, buying and comments/assessment of satisfaction after a transaction. The question we try to answer by observation of sellers' and customers' behaviour is what the most efficient techniques and auction features that may lead to successful transaction are. The importance of risen problem comes from the fact that sellers may choose various paid marketing tricks and add several promotional features to on-line auctions that may focus visitors' attention on particular product and auction but there may also be circumstances that increase or reduce probability of successful transaction which cost nothing (or are relatively cheap) but come from selling experience and good practices. Additional paid marketing features finally increase product price (which may result in lower number of transactions) or reduce seller's profit.

Conclusions drawn from our experiments may be red from two perspectives: seller's and buyer's one. While we try to identify auction features that increase the chance of selling a product the ranking red in opposite direction may show the attributes of auctions that are especially interesting for customers because of low number of bids which may lead to lower price of purchased product.

We performed our experiment between February and May 2009 collecting data from on-line auction server Allegro.pl. *Allegro* is a leading on-line auction service in Poland and is also available (under different brands) in several countries in Europe. However, to focus on Polish market only we filtered and removed data describing foreign offers from our database after collection process. During data collection phase we gained description of auctions with IDs from 550459579 to 6024299121. It resulted in storing information on 36 million auctions, 18 million customers' offers and 150 million links between an auction and product categories where particular auction was catalogued. We decided to collect for analysis auctions that ended during one particular month. Because more that 60% of auctions collected in our experiment ended in March 2009 we took this set of data for further analysis. After this filtering we had 11 835 679 auctions ending in March 2009 to analyse.

According to aukcjostat.pl, currently (in June 2012), more that 71% on on-line auctions in Poland are presented in Allegro.pl. Number of auctions is systematically growing and in the end of June 2012 offer counter shows above 21 million of running auctions (see fig 1.). Huge numbers and popularity of Allegro.pl are key reasons why it is worth to analyze if sellers' behaviour is rationale and how one can define good practices that lead to savings on paid marketing features.

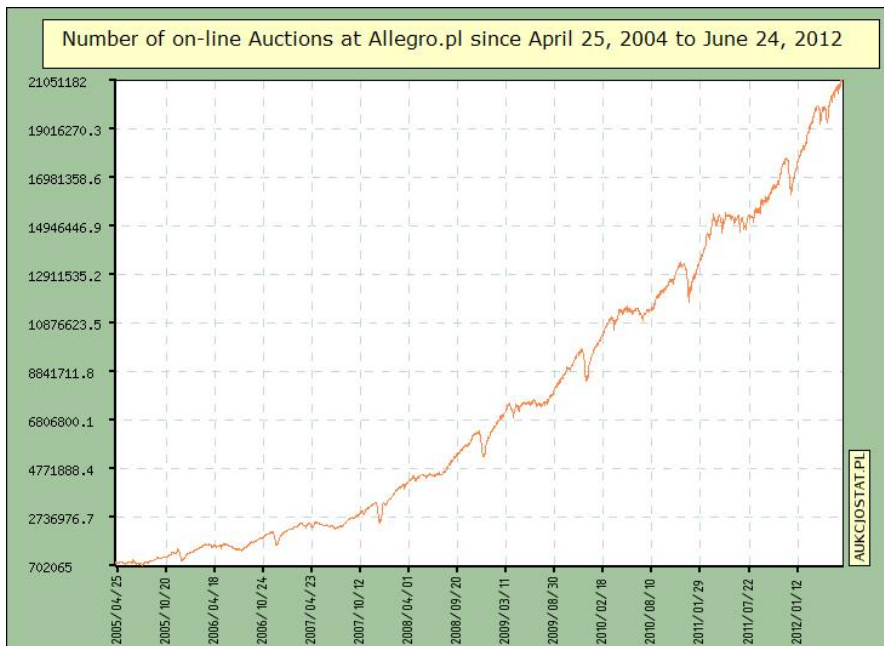


Fig. 1 Number of auctions on Allegro.pl from Apr. 2005 to Jun. 2012. Source: aukcjostat.pl

For comparison the strongest *Allegro*'s competitor in on-line auctions in Poland – Aukcysz.pl has 10.5% part of the market and eBay Poland share is 1.3%, according to statistics provided by aukcjostat.pl in June 2012.

3 Analysis of Collected Data

Within the set of collected data cleared and filtered records related to auctions ending in March 2009 taken into account in further analysis are the following:

- 11 835 679 auctions
- 9 329 736 bids in auctions
- 52 469 216 records describing links between an auction and product category (or subcategory) in *Allegro* system.

For analysis purpose we define the following terms:

- **Auction-transaction** – at least one product offered on particular on-line auction was sold.
- **Product category** – each auction registered in *Allegro* system may be catalogued in many product subcategories, however each product subcategory must belong to one of 23 main categories displayed on home page of *Allegro* auction system. Thus we assume that each product offered on an auction belongs to one of 23 main categories.

- **Effectiveness of sale** – what part (percent) of auctions from chosen set was finished with transaction.

3.1 Products in Categories

The first perspective to analyse collected data is popularity of product categories. One may ask a question if *Allegro* system as an on-line auction platform is especially convenient place to trade relatively cheap and small products – easy to deliver by postal service. On the other hand on-line auction system may be a place where consumers look for expensive products to wider competition of offers compared to local stores and benefit in noticeable savings or big, heavy goods which the consumers are not willing to transport home from local stores by their own.

Table 1 Distribution of auctions belonging to particular product categories, number of auctions ended with transaction and effectiveness of sale in all main product categories

Product category	Number of auctions	Auction-transactions	Effectiveness of sale [%]
Antiques and Art	300988	76294	25.35
Jewellery and Watches	392893	50427	12.83
For Children	1171594	271031	23.13
House and Garden	878568	53264	6.06
Films	152089	26616	17.50
Business and Industry	272422	10496	3.85
Photography	101777	14366	14.12
Games	156889	42805	27.28
Collections	611983	209210	34.19
Computers	338335	51555	15.24
Books and Comics	959583	151468	15.78
Cars and Motors	2369319	111179	4.69
Music & Instruments	306986	62900	20.49
Real Estate	12541	92	0.73
Clothes & Shoes	2237657	474461	21.20
Work	0	0	-
Electronics & Household	352090	44199	12.55
Sport & Travel	425146	55568	13.07
Telephones	277134	50683	18.29
Holidays	34386	84	0.24
Health & Beauty	427882	44689	10.44
Erotic	0	0	-
Other	55417	3124	5.64

In table 1 we present numerical data describing distribution of auctions belonging to particular product categories, number of auctions ended with transaction and effectiveness of sale in all main product categories. Two categories are empty (*Work* and *Erotic*). Category *Work* was filled with announcements, which were filtered out at data preparation stage, while category *Erotic* was introduced to *Allegro* system on May 21, 2009 – after our experiment. Low effectiveness of sale in category *Real Estate* comes from the fact that those transactions cannot be legally finalized in virtual environment in Poland but require a notarial act. This may lead to conclusion that some of auctions belonging to *Real Estate* category could be finalized outside on-line auction system. However this is only a supposition not proved in our experiment.

To understand collected data one should notice that higher number of auctions in particular product category means higher number of products and wider price competition which may result in growing attractiveness of on-line auction site assessed from customer's perspective. Five the most popular product categories were:

- Cars & Motors
- Clothes & Shoes
- For Children
- Books & Comics
- House & Garden

Almost 2/3 (64.35%) of total number of auctions available in *Allegro* system were indexed in five top counted (out of 23) product categories. 58.82% of all auctions ended with transactions were catalogued in 5 top counted product categories. It is worth to remind that in auction-transaction we do not count number of products sold but number of auctions where at least one product was sold. Thus the actual number of transactions (or products sold) had to be higher, because many auctions are multi-item ones.

Five top-selling product categories, where effectiveness of sale reached the highest level, were:

- Collections (34.19%)
- Games (27.28%)
- Antiques & Art (25.35%)
- For Children (23.13%)
- Clothes & Shoes (21.20%)

3.2 Time of Auction Close May Affect Probability of Transaction

Lifetime of an offer presented in on-line auction system is subject of research works, e.g. [1]. Pretty often one can observe that the highest attention of bidders is addressed in last minutes or even last seconds before auction close. This is the time when potential buyers try to win the auction with lowest possible price. It also became a practice, especially in fixed time auction systems, that software agents play the role of auctions snipers to send bids in last moments before auction

closes. Depending on auction site policy auction sniping may be allowed or forbidden, e.g. eBay Germany banned auction sniping [2] which practice was declared illegal by Berlin's County Court [3]. One of practices used by some auction sites (e.g. *iGavelAuctions.com*, *TradeMe.co.nz*) to mitigate inequality of chances between software agent auction snipers and bidding humans is extending the bid deadline when a new bid is placed in last moments before auction end.

Daily perspective

Day of week when an auction is ending may have an influence on probability of transaction. It may come from the fact that the more visitors display and read auction offers the more likely it is that someone buys a product. To measure the importance of termination-day-of-week factor on successful transaction we calculated efficiency of sale for all auctions terminated on particular day of week in analysed period. Results are collected in table 2.

Table 2 Day of week when auction is terminated vs. probability of transaction

Day of week when auction terminates	Effectiveness of sale [%]
Monday	14.89
Tuesday	14.29
Wednesday	15.44
Thursday	15.22
Friday	14.59
Saturday	17.72
Sunday	19.46

Analysis show that there was a noticeable difference in sale effectiveness on particular days of week. While on Saturdays and Sundays chance to sell a product on an auction was 17.72% and 19.46% respectively on the least trading day - Tuesday only 14.29% of auctions ended with transaction.

Hourly perspective

Activity of auctioneers differs in various periods round the clock. One may expect that during night hours less buyers manually review offers and sends bids. But how significant is this time influence? In order to assess the effect of auction termination timing on efficiency of auction-transaction we made analysis collected in table 3. We checked efficiency of sale against auction ending hour – what percentage of auctions terminating within particular hour was ended with trade transaction.

Experiments showed clearly that chances to sell a product on an on-line auction depend on auction termination time. The highest ratio of successful auction-transaction against total number of auctions ending within particular hour occurred between 19:00 and 22:00 (evening hours). Observed relation confirms our earlier

Table 3 Efficiency of trade on on-line auctions on auction termination time scale

Time of auction termination	Efficiency of sale [%]	Time of auction termination	Efficiency of sale [%]
00:00 – 00:59	9.95	12:00 – 12:59	12.58
01:00 – 01:59	9.20	13:00 – 13:59	12.59
02:00 – 02:59	8.85	14:00 – 14:59	12.90
03:00 – 03:59	8.29	15:00 – 15:59	12.99
04:00 – 04:59	3.09	16:00 – 16:59	14.52
05:00 – 05:59	6.22	17:00 – 17:59	16.62
06:00 – 06:59	8.10	18:00 – 18:59	18.62
07:00 – 07:59	8.61	19:00 – 19:59	20.55
08:00 – 08:59	8.68	20:00 – 20:59	22.19
09:00 – 09:59	9.66	21:00 – 21:59	21.28
10:00 – 10:59	10.79	22:00 – 22:59	16.41
11:00 – 11:59	11.94	23:00 – 23:59	12.55

speculations that the highest bidders activity occurs on last minutes of auction time. Another conclusion we can draw from collected results is that huge part of auctions is bided manually, because transactions are fixed in periods when customers have free time to enjoy on-line auction shopping.

Late night hours (or as one might say very early morning hours), especially from 4:00 to 5:00 is the worst period for auction ending from seller perspective. The probability of transaction in auctions terminating between 4 and 5 a.m. reached 3%, which was seven times less when compared to peak trading (evening) hours.

Lesson learned from above experiment is that deep late night/very early morning is the best time for bargains in on-line auctions and sellers should tend to terminate their timed auctions in evening hours when activity of bidders reaches the highest level.

3.3 Paid Features to Promote an Offer on Auction Site

Allegro system offers several paid options to promote auctions. They include **miniature** product photo displayed next to auction title on auction list, **high-lighted** (yellow background) display on auction list and among search results, **bold** font on auction list, **priority** (priority auctions are listed before ordinary auctions), advertising on product **category** auction list, advertising on **main page** of auction site. Regardless from effect (whether an auction ends with transaction or not) the seller covers extra cost when s/he decides to apply those features to promote her/his auction.

To assess how far paid promotional features may grow customers' interest in offered product we computed average number of auction display (views, visits) for auctions where particular sets of paid promotional features were applied. Result of

Table 4 Popularity of paid promotional auction features and average number of displays/visits on advertised auctions

Type of paid advertisement features	Number of auctions where advertisement was applied	Average number of displays(visits) on advertised auction
miniature	9563217	18.3
no paid advertisement	1420171	6.7
miniature + bold	484152	419.2
miniature+bold+priority	94730	74.2
miniature+highlight+bold+priority	43971	724.1
miniature+highlight+priority	21519	619.6
miniature+highligh	19751	79.6
miniature+highlight+bold	14637	112.6
bold	9080	37.8
bold+priority	4263	107.6
priority	4070	139.7
highlight+bold+priority	1362	169.8
highlight+bold	1017	41.4
highlight	805	55.4
miniature+highlight+bold+priority+main page	595	3777.9
miniature+priority+main page	470	4297.6
miniature+bold+priority+main page	458	40006.2
highlight+priority	434	93.3
miniature+highlight+bold+priority+category+main page	408	4324.6
miniature+highlight+bold+priority+category	326	1312.5
miniature+main page	153	3274.0
miniature+category	130	623.7
miniature+priority+category+main page	107	3810.5
Miniature+bold+priority+category+main page	100	4014.7

our analysis is presented in table 4. We reduced the list to sets of feature combinations which had at least 100 instances (auctions with applied features) within observed period.

Statistics collected in table 4 show that additional promotional features increase customer's interest and auction page is visited more frequently when it is advertised. On-line auctions without any advertising features were visited 6.7 times in the average. Then we checked what part of all auctions had particular advertising features applied and how those options increased visits counter when compared to auctions without particular feature. We also analysed price effectiveness of buying advertising features. Results are collected in table 5.

Table 5 Popularity of advertising features, how they grow number of visits on auction page and effectiveness of paid promotion of an auction

A	B	C	D	E
Advertising feature	What part of all auctions applied the feature [%]	Feature price [PLN] 1 PLN = US\$ 0.30	Generated growth of auction displays [%]	Growth of display per 1PLN [%] (D/C)
Miniature	87.82	0.15	247	1646.6
Highlight	0.89	6.00	61	10.2
Bold	2.71	2.00	50	25.0
Priority	5.55	12.00	438	36.5
Category page	0.01	29.00	417	14.4
Main page	0,02	99.00	2341	23.6

The survey collected in table 5 shows that miniature image displayed by auction title was the cheapest and most price effective form of advertising an auction. Thus it is not a surprise that it is also the most popular promotional feature applied to 87.82% of auctions. The winner advertising feature in growth of visits was advertising on auction site's main page (visit counter increased by 2341% when compared to auctions without this feature), however high price of this promotion may pay back only if customers' visits turn into transactions. Considering price effectiveness of promotional auction features priority seems to be a reasonable option which balance cost of 12 PLN which seller must pay regardless auction result with 438% growth of customers' visits. Priority, after a miniature, is the second most popular advertising feature, applied to 5.55% of auctions.

4 Summary

In our research we tried to observe sellers' and customers' behaviour on o-line auctions. Collected data provide a quantitative arguments in discussion on effectiveness of promotional features used in on-line auctions. Experiment was conducted on leading Polish auction site *Allegro* in May 2009 and effectiveness analysis may differ when compared to other auction sites, because of different pricing of auction's promotional features.

One should remember that number of visits on an auction page may not easily convert into transactions. Internet auction sites are very competitive marketplace and paid advertising features are only a part of marketing tricks that may influence sells. From customer's perspective it is good to be aware of adverting methods and avoid emotions when buying goods.

Main conclusions drawn from our experiment are the following:

- Almost 2/3 (64.35%) of total number of observed auctions were indexed in five top counted (out of 23) product categories.

- There was a noticeable difference in sale effectiveness on particular days of week. On Saturdays and Sundays chance to terminate an auction with transaction was 17.72% and 19.46% respectively while on Tuesdays only 14.29% of auctions ended with transaction.
- Experiments showed that the highest ratio of successful auction-transaction against total number of auctions ending within particular hour occurred between 19:00 and 22:00 (evening hours).
- Among paid promotional features advertising on auction site's main page was the most effective in increasing visits/displays (by 2341%), however high price of this promotion may pay back only if customers' visits turn into transactions.
- A miniature image displayed by auction title was the cheapest and most price effective form of advertising an auction.

References

1. Yang, I., Kahng, B.: Bidding process in online auctions and winning strategy: Rate equation approach. *Physical Review E* 73(6), 67101 (2006)
2. Steiner, D.: eBay Germany Bans 'Sniping' Services, *EcommerceBytes.com* (October 25, 2002)
3. Sniper-Software doch legal (September 25, 2003), <http://heise.de/-85845> (in German)
4. Wojciechowski, A., Musial, J.: A Customer Assistance System: Optimizing Basket Cost. *Foundations of Computing and Decision Sciences* 34(1) (2009)
5. Yang, Y., Wang, C.: Recent Development in Online Auction Research: A Literature Review. In: *Proceedings of Business and Information*, vol. 7 (2010)
6. Warczynski, P.: Effectiveness Analysis of Internet Auctions Participants' Behaviour. Master thesis under supervision of Wojciechowski A. Poznan University of Technology, Poznan, Poland (2009) (in Polish)

Part VII
Ph.D. Consortium

Data Mining Approach to Digital Image Processing in Old Painting Restoration

Joanna Gancarczyk

Abstract. In this paper an attempt has been made to apply data mining techniques to the task of separation and categorization features in digital images of artworks. Both craquelure separation and retouching identification are important steps in art restoration process. Since the main goal is to enable recognition of character and cause of damage, as well as forecasting its further enlargement, a proper tool for precise detection of the pattern is needed. However, the complex nature of the pattern is a reason why a simple, universal detection algorithm is not always possible to implement. Algorithms presented in this work apply mining structures which depend of expandable set of attributes forming a feature vector, and thus offer an elastic structure for analysis¹

1 Introduction

Last decades brought a significant improvement of availability and effectiveness of digital imaging tools applied to various disciplines. Therefore also art analysis technique has adopted much of their potential, enabling faster, wider and more exhaustive research. In [19] Stork indicates brush stroke and craquelure identification, dewarping, perspective and lighting analysis as most exploited and promising up to date fields of investigation. Other authors mention also research areas like recombination of fragments, virtual restoration and lacuna filling (see [7], [5], [8] and [18] for summary analysis). Originally most of the task were resolved by means of well defined image processing operations like thresholding, filtering and mathematical morphology. Data mining techniques applied together with the above mentioned

Joanna Gancarczyk
University of Bielsko-Biala, Willowa 2, 43-309, Bielsko-Biala, Poland
e-mail: jgan@ath.bielsko.pl

¹ This work was partially supported by NCN (National Science Centre) under grant no. 6593/B/T02/2011/40.

methods is a novel approach. Here models for analysis are based on decision trees and data clustering algorithms.

2 Virtual Restoration

Virtual restoration is a non-invasive method of digital modeling of restoration work. As tools for artwork restoration, image processing techniques serve two purposes. They can be used as a guide for the actual restoration of the artwork (computer-guided restoration) or, alternatively, they can produce a digitally restored version of the work (virtual restoration) [5].

In the class of methods for virtual artwork restoration, we can include the algorithms for removing cracks from paintings and frescos. Cracks are often caused by a rapid loss of water in the paintings varnish. When the painting itself is located in a dry environment, a nonuniform contraction of the varnish covering can cause the birth of cracks. With image processing tools it is possible to entirely remove cracks by means of interpolation techniques ([12], [11]). An algorithm for crack removal is usually a two-step procedure: first, the cracks have to be detected; next, the selected cracks are filled in. The crack selection step can be semi-automatic, or automatic.

Another technique belonging to the class of methods for virtual artwork restoration is lacuna filling. Lacunas (missing paint in small areas) are a common form of damage that can occur to paintings. The damaged areas have first to be identified and then filled to give the impression of continuity of the image. In particular, some restoration methods have been implemented by basically referring to two different restoration schools: the Scuola Romana of Cesare Brandi and the Scuola Fiorentina of Umberto Baldini. According to these restoration schools a lacuna can be distinguished as restorable or non-restorable: in the former case, the lacuna still contains some original colours or a part of the original drawing (called sinopia); in the latter, the lacuna consists of a large damaged area of a painting where it is not possible to recover the original drawing [7]. Identification of lacunas is generally a task of filtering and region growing operations.

3 Image Processing Methods

3.1 *Thresholding and Segmentation*

There are three general approaches to segmentation, termed thresholding, edge-based methods and region-based methods. In thresholding, pixels are allocated to categories according to the range of values in which a pixel lies. In edge-based segmentation, an edge filter is applied to the image, pixels are classified as edge or non-edge depending on the filter output, and pixels which are not separated by an edge are allocated to the same category. Region-based segmentation algorithms

operate iteratively by grouping together pixels which are neighbours and have similar values and splitting groups of pixels which are dissimilar in value.

3.2 *Mathematical Morphology*

Mathematical Morphology is a tool for extracting image components that are useful for representation and description. The technique was originally developed by Matheron and Serra [17]. It is a set-theoretic method of image analysis providing a description of geometrical structures. Morphology can provide boundaries of objects, their skeletons, and their convex hulls. It is also useful for many pre- and post-processing techniques, especially in edge thinning and pruning.

Most morphological operations are based on simple expanding (dilation) and shrinking (erosion) operations. The primary application of morphology occurs in binary images, though it is also used on grey level images. Erosion and dilation can be used in a variety of ways, in parallel and series, to give other transformations including thickening, thinning, skeletonisation and many others and their result depends of a structuring element. Two very important transformations are opening (dilation of the erosion of a set A by a structuring element B) and closing (erosion over dilation). Opening removes small objects from the foreground (usually taken as the dark pixels) of an image, placing them in the background, while closing removes small holes in the foreground. The top-hat and bottom-hat transforms - subtracting the original image from closing and opening from the original image respectively - is used for extracting small features in an image. It is useful when variations in the background mean that this cannot be achieved by a simple threshold.

The fundamentals of image processing methods may be found in [16] and [10].

4 **Lacuna and Retouching Identification**

Both lacunas and retouchings form flat shapes on the surface of a painting. In a digital image processing the usual way to separate them from the original paint layer is to define a set of initial points and then to apply some region growing algorithm on the basis of color and texture of a surrounding area [7]. In [13] the Bayesian classifier is applied to perform the rough lacuna regions detection. Authors of [18] refer also to the methods based on basic point operations, mathematical morphology and spatial filtering.

It is a common practice to perform the lacuna and retouching extraction task on images taken under ultraviolet light. UV fluorescence photography can reveal the presence of natural resin varnishes, as these often fluoresce under UV light while newer varnishes do not. It is also possible to identify any retouchings and over-paintings as they appear as nonfluorescing dark spots in contrast to the original fluorescent areas [14]. Figure 1 presents an UV photography of a 19th century Rafał Hadziewicz work "The Holly Family", property of National Museum in Krakow.



Fig. 1 UV fluorescence photography of Rafał Hadziewicz work "The Holly Family" with a selected fragment for further analysis

4.1 Clustering Method

There are two situations in which manually defining an initial set of points for retouching identification is not suitable. One is when the cardinality of separated retouched fragments is too high, and the ratio of work input by the restorer to the final result is not adequate. Second is if not only does the shape have a meaning, but also the shade of the retouchings' image. That is, when there are retouchings originating from different periods and performed in different techniques.

The image clustering method presented in this work lets the restorer apply a fast algorithm and get a satisfactory result automatically for the beginning, then letting him achieve a final separation result by thresholding the clustered image with a chosen threshold level. The case presented below (Figure 2) is based on color and brightness analysis of a selected fragment of the UV image. Feature vectors related to each pixel are built of its red, green, and blue channel values as well as on the grayscale value. Methodology of the proposed solution is based on three steps: (1) reading the matrix of image pixel values, (2) generating feature vectors for a data mining model, (3) analyzing the achieved splitting rules, (4) thresholding the initial image with respect to the rules. This path should be supplemented in further research by an additional image processing step, based on filtering and mathematical morphology tools, so that the feature vector would be completed with more attributes to improve the results.

The Microsoft Clustering algorithm, which was applied during the research, provides two methods for creating clusters and assigning data points to the clusters. The first, the K-means algorithm, is a hard clustering method. This means that a data point can belong to only one cluster, and that a single probability is calculated for the membership of each data point in that cluster. The second method, the

Expectation Maximization (EM) method, is a soft clustering method. This means that a data point always belongs to multiple clusters, and that a probability is calculated for each combination of data point and cluster [15]. In the presented example the default EM method was used with the expected number of clusters manually set to three. This value was chosen by a subjective, visual judgement.

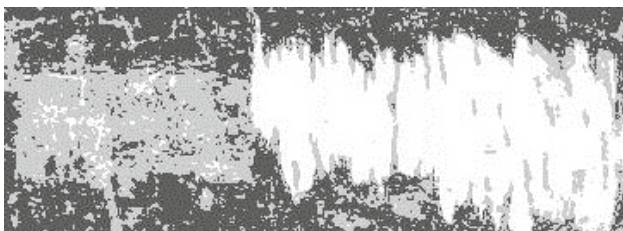


Fig. 2 Two classes of retouchings recognized by the clustering algorithm

4.2 Result

The number of classes given in the construction of a mining model refers to two retouched areas seen on the investigated fragment and the background. The number of missclassified pixels is acceptable for a rough analysis of size and shape of the retouchings. Further improvement of the result might be achieved by another pre- and post-processing of the image, including denoising, smoothening, sharpening and edge detection.

5 Craquelure Separation

Craquelure is a pattern of cracks that appears in a painting during the process of aging. Cracks in the paint layer grow as the canvas or wood support of the painting moves in response to changes of humidity and temperature. Every layer of a painting has its own distinctive mechanical behaviour, and therefore, every layer contributes in its own way to the formation of craquelure ([20], [21]).

The semi-automatic method of craquelure separation described in [4] and then recalled in [7] is based on a manual selection of at least one starting point for each separated piece of a craquelure pattern. Then an iterative process is run to expand the structure according to the gradient of pixel values. This approach is adequate due to the character of a craquelure pattern which is formed of linear, continuous shapes of changing shade, but extinguishable from the background. In automatic selection model cracks are identified by means of a proper filter, like Gabor filters, or a morphological filter called top-hat transform ([1], [2], [3]). However, with this approach not only cracks, but also brush strokes and other texture characteristic could be detected. This problem can be solved by discriminating brush strokes and

cracks on the basis of shape, hue or saturation values or thickness and favoured orientation.

5.1 Decision Tree Approach - Training Data and Feature Vector Generation

The example of craquelure pattern separation by means of a decision tree construction presented in this paper is a continuation of a method wider described in [9]. The main goal is to resign from manually indicating initial points of craquelure pattern, as it was proposed by Barni et.al [4]. Instead, these points are chosen automatically according to the set of rules obtained in a mining model.

The manual step is not omitted completely, since it remains the most precise way to define craquelure pixels in a training set for the mining model. However, the ratio of manual work applied by the restorer to the obtained result might be acceptable, and in many cases more adequate then in the original method. The training set is defined on a small area selected as a representative region of the whole image. In case the painting is not consistent, that means the parts differ from one another significantly a few training areas may be chosen concurrently according to the colour, brightness and texture. See Figure 3 to observe the training points selection for the analysed fragment of 19th century Rafał Hadziewicz painting "Portrait of Wentzl".

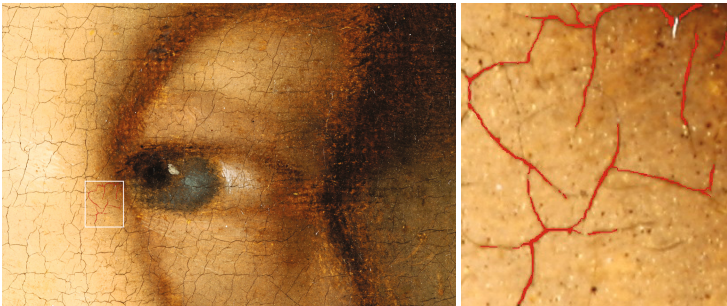


Fig. 3 Training set definition for a decision tree

Once the training set is defined and craquelure pixels selected a binary mask is created for the training area. See Figure 4(left) to compare.

A mining structure was defined with application of the Microsoft Decision Trees model. A source table consisted of feature vectors for each pixel in the training set area (R,G and B channel values, grayscale value, median filtered image value and the difference between median filtered and original image) and class labels referring to the mask. There are two class labels 1 and 0 for craquelure and non-craquelure pixels respectively.

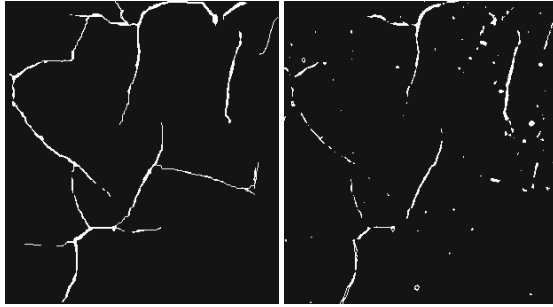


Fig. 4 Mask and the computed initial set of craquelure pixels for the same area



Fig. 5 Initial set of craquelure pixels for the whole image (negative)

5.2 Results

The mining model applied in the crack selection task distinguished three significant classes of pixels that may be assumed to be craquelure pixels. The classes are based on (1) - red value, (2) - grayscale value, (3) - median filtered value and (4) - difference between grayscale value and median filtered image. That means four parameters out of six appeared to me meaningful while performing the computations.

Figure 4 presents the mask generated in the training data preparation step compared with a new mask based upon the rules obtained by building the decision tree. The latter is intended to be an initial set for craquelure pattern detection by means of a region growing algorithm. Several missclassification may be observed in the right side picture. This is partially due to the character of the image - no smoothing nor noise reduction process was applied before the actual computation. Also the set of parameters of vector features would be a subject of further research. Best results, comparing to the manual method, are achieved when the crack pattern is inconsistent

(not solid) and the background relatively homogeneous. That is because then only a small representative area is necessary to define the training set. Figure 5 presents the initial set of craquelure pixels for the whole image, obtained by application of the achieved rules.

6 Conclusions and Further Work

A new technique of craquelure and retouching identification was presented. The novel approach is based on application of mining models to the analysis. This enabled some automatisation in a process of defining initial points of craquelure and rough detection of retouched areas. Though none of these methods can replace the restorer's work completely, a significant help is obtained, thus letting more caution to be paid to further steps of the investigation on the artwork. The attributes for above methods are based on the colour of particular pixels, grayscale value and grayscale value after median filtering. Further work will concern better adjustment of the feature vector to obtain more suitable classification results.

Acknowledgements. Author would like to thank Mrs. Joanna Sobczyk and the Laboratory of Analysis and Nondestructive Investigation of Heritage Objects of the National Museum in Krakow for substantial support and making accessible high resolution images of paintings from the collection of Rafał Hadziewicz works.

References

1. Abas, F.S.: Analysis of Craquelure Patterns for Content-Based Retrieval. PhD Thesis, University of Southampton, Southampton (2004)
2. Abas, F.S., Martinez, K.: Classification of painting cracks for content-based analysis. In: IST/SPIE's 15th Annual Symp. Electronic Imaging, Santa Clara, California, USA (2003)
3. Abas, F.S., Martinez, K.: Craquelure analysis for content-based retrieval. In: Proc. of 14th Int. Conf. on Dig. Sig. Proc., Santorini, Greece, pp. 111–114 (2002)
4. Barni, M., Bartolini, F., Cappellini, V.: Image processing for virtual restoration of artworks. *IEEE Multimedia* 7(2), 34–37 (2000)
5. Barni, M., Pelagotti, A., Piva, A.: Image processing for the analysis and conservation of paintings: opportunities and challenges. *IEEE Sig. Proc. Mag.* 141 (2005)
6. Bucklow, S.L.: A sylometric analysis of Craquelure. *Computers and the Humanities* 31, 503–521 (1998)
7. Cappellini, V., Barni, M., Corsini, M., de Rosa, A., Piva, A.: ArtShop: an art-oriented image-processing tool for cultural heritage applications. *J. Visual Comput. Animat.* 14, 149–158 (2003)
8. Cappellini, V., Piva, A.: Opportunities and Issues of image processing for cultural heritage applications. In: Proc. EUSIPCO 2006, Florence, Italy (2006)
9. Gancarczyk, J.: Decision tree based approach for craquelure identification in old paintings. *AISC* (in press)
10. Gonzalez, R.C., Woods, R.: *Digital Image Processing*, 3rd edn. Prentice Hall (2007)
11. Gupta, A., Khandelwal, V., Gupta, A., Srivastava, M.C.: Image processing methods for the restoration of digitized paintings. *Thammasat Int. J. Sc. Tech.* 13(3), 66–72 (2008)

12. Hanbury, A., Kammerer, P., Zolda, E.: Painting crack elimination using viscous morphological reconstruction. In: Proc. ICIAP 2003, Mantova, Italy (2003)
13. Liu, J., Lu, D.: Knowledge Based Lacunas Detection and Segmentation for Ancient Paintings. In: Wyeld, T.G., Kenderdine, S., Docherty, M. (eds.) VSMM 2007. LNCS, vol. 4820, pp. 121–131. Springer, Heidelberg (2008)
14. Lizun, D.: Fine Art Conservation, <http://fineartconservation.ie/damian-lizun-fine-art-conservation-4-4-43.html>
15. Microsoft Clustering Algorithm Technical Reference, <http://msdn.microsoft.com/en-us/library/cc280445>
16. Tadeusiewicz, R., Korohoda, P.: Computer Analysis and Image Processing (in Polish: Komputerowa analiza i przetwarzanie obrazow). Progress of Telecommunication Foundation Publishing House, Krakow (1997)
17. Serra, J.: Image Analysis and Mathematical Morphology, vol. I. Ac. Press, London (1982)
18. Sobczyk, J., Obara, B., Fraczek, P., Sobczyk, J.: Zastosowania analizy obrazu w nieniszczących badaniach obiektów zabytkowych. Wybrane Przykłady, Ochrona Zabytków 2, 69–78 (2006)
19. Stork, D.G.: Computer vision and computer graphics analysis of paintings and drawings: An introduction to the literature. In: Proc 13th Int. Conf. on Computer Analysis of Images and Patterns, pp. 9–24 (2009)
20. Stout, G.L.: A trial index of laminal disruption. JAIC 17(1, 3), 17–26 (1977)
21. De Willigen, P.: A Mathematical Study on Craquelure and other Mechanical Damage in Paintings. Delft University Press, Delft (1999)

Determining Document's Semantic Orientation Using k NN Algorithm

Krzysztof Jędrzejewski and Maurycy Zamorski

Abstract. This paper presents another approach for determining document's semantic orientation process. At first there is brief introduction describing the area of application of opinion mining, and some definitions to help the reader better understand later text. Then there are mentioned most commonly used methods and an alternative one described in Section 5. At the end there are experiment results showing that k NN algorithm is giving similar results to proportional algorithm proposed in [5].

Keywords: opinion mining, text mining, semantic orientation.

1 Introduction

As global Internet network grows rapidly, it is commonly used by vast number of people to exchange information. By information we can treat almost anything, from newspaper articles, to video streaming. One of quite new phenomena is an advent of social network websites, discussion boards (forums), price and product comparators and much more, where users can share their opinions in certain areas. Many such pages implement mechanisms of valuation, where one can apart of writing a comment, choose whether this comment is positive or negative – in simplest case. Problem appears when we deal with text only, without any additional information on the character of the statement, e.g. on discussion boards, or in raw comments to some newspaper article. In this situation an only solution is to process the text, preserving the semantics of an expression in such way, that it can be understood by a computer algorithm. After that, we can evaluate, with a certain probability, whether processed phrase has a positive or negative value and therefore classify it to a positive or negative class of an examined data collection.

Krzysztof Jędrzejewski · Maurycy Zamorski
Institute of Computing Science, Piotrowo 2, 60-965 Poznan, Poland
e-mail: Krzysztof.Jedrzejewski@cs.put.poznan.pl
Maurycy.Zamorski@cs.put.poznan.pl

Such knowledge is desirable not only for pure scientific reasons, but has a great value in marketing. By having opinions data changing in some period of time we can evaluate for example a lifecycle of a certain product, and thus predict income and profitability of investment. Also such data can be used to upgrade certain weaknesses resulting in better quality and therefore larger profit.

Other areas where this kind of data is useful are e.g. political sciences. We can gain much information from opinions on certain topics and treat them as a part of social consultations process and even more. By continuous monitoring of public opinion on certain topics we can implement complex social systems with feedback loop, concerning e.g. law regulations or such.

2 Related Work

Determining document's semantic orientation is one of tasks performed in area of opinion mining. Opinion mining is a new scientific domain, strongly related to data mining, and also to natural language processing or machine learning. Opinion mining methods can be approached in several different ways:

- supervised learning methods [2,3] – methods using training sets as a starting point for comparison and evaluation of performance;
- unsupervised learning methods [4,9] – methods without training set, mainly used for grouping objects and cluster analysis;
- information retrieval methods [7,14] – methods of retrieving information from documents in the full sense.

In this paper we strongly relate to scoring algorithm described in [5], where proportional method was used in process of determining document's semantic orientation. That method is being based on another approach described in [2], which we refer to as score method.

There are also other works concerning opinion mining [2,13] describing conceptions to deal with documents modeled as sets of words or vectors, and also many more [6].

3 Main Definitions

Before we get to k NN method description there is a need to explain some definitions, crucial in understanding entire concept:

- document – whole examined set of words, e.g. a statement or a collection of statements on certain topic;
- term – a single word, being a component of a document; term may be also after lemmatization or stemming – a processes of identifying core of a word, without any grammatical changes;
- n-gram – a part of a text containing n characters, including white spaces;
- token – used interchangeably, means a word or an n-gram;

- training set – set of raw, unchanged text documents with classes assigned arbitrarily, in most cases by a human;
- positive / negative class – describes whether a document contains a positive or negative option on certain topic;

4 Basic Concepts

Document’s semantic orientation is often calculated by aggregation of the level of relationship between a words found in this document and a positive or negative class.

$$\gamma(d) = \begin{cases} C_P, & eval(d) > 0 \\ C_N, & eval(d) < 0 \end{cases} \tag{1}$$

where

$$eval(d) = \frac{\sum_{t_i \in d} score(t_i)}{|d|} \tag{2}$$

or

$$eval(d) = \sum_{t_i \in d} score(t_i) \tag{3}$$

where t_i is the i -th term of the document d , $|d|$ is the number of terms in a document d , C_P and C_N are positive and negative classes, $score()$ is a function assigning positive or negative values to terms, depending on their relation with the suitable class.

One of the methods utilizing concept of supervised learning [2] has a score function in the following form:

$$score(t) = \frac{p(t|C_P) - p(t|C_N)}{p(t|C_P) + p(t|C_N)} \tag{4}$$

where $p(t|C_P)$ and $p(t|C_N)$ are conditional probabilities of the occurrence of the term t in positive and negative class, respectively. These probabilities can be approximated by term occurrence frequencies in the training set.

Different approach based on pointwise algorithm was introduced in [5], which is based on the ratio of term occurrence frequency in documents assigned to a positive and negative classes. The score function is as follows:

$$score(t) = \begin{cases} p_t - 1 & , \text{ iff } p_i \geq 1 \\ -\left(\frac{1}{p_t} - 1\right) & , \text{ iff } p_i < 1 \end{cases} \tag{5}$$

where

$$p_t = \frac{p(t|C_P) + \epsilon}{p(t|C_N) + \epsilon} \tag{6}$$

where, p_t is the raw semantic orientation of the term t , $p(t|C_P)$ and $p(t|C_N)$ are conditional probabilities of occurrences of the term t in documents from positive

and negative classes, respectively, and ε is a small positive value controlling for terms that appear in only one class.

To the method of calculation described by formula (4) we refer to as score method, and to the one described by (5) we refer to as proportional method.

5 Our Approach

The method proposed in this paper for determining document's semantic orientation is use of k -nearest neighbor (k NN) algorithm, that assigns classified object to the class most common in the collection of k examples from training set \mathcal{E} , that are most similar:

$$\gamma(d) = \begin{cases} C_P, & \text{iff } |SIM_k(d) \cap C_P| > |SIM_k(d) \cap C_N| \\ C_N, & \text{iff } |SIM_k(d) \cap C_P| < |SIM_k(d) \cap C_N| \end{cases} \quad (7)$$

where γ is semantic orientation of a document d , C_P and C_N are positive and negative classes, respectively, $SIM_k(d)$ is the nearest neighborhood of document d , i.e. training set's subset, which cardinality is equal k , and

$$\forall e \in SIM_k(d) \quad \forall e' \in \mathcal{E} \setminus SIM_k(d) \quad sim(d, e) \geq sim(d, e') \quad (8)$$

where \mathcal{E} is a training set and $sim(d, e)$ denotes a function measuring similarity between documents d and e .

In addition to the classic version of k NN algorithm, we propose in this paper two concepts inspired by this approach.

The first algorithm, Amplified Neighbors' Similarities Sum (NSS), takes into account level of similarity between classified document and examples in its nearest neighborhood in \mathcal{E} and disproportion between number of examples representing positive and negative class in the training set.

$$\gamma(d) = \begin{cases} C_P, & \text{iff } eval(d) > 0 \\ C_N, & \text{iff } eval(d) < 0 \end{cases} \quad (9)$$

where C_P and C_N are positive and negative classes, respectively, and $eval(d)$ is a function evaluating document, expressed by the formula:

$$eval(d) = \sum_{e \in SIM_k(d)} (sim(d, e) \times pol(e) \times disp(e)) \quad (10)$$

where $sim(d, e)$ denotes a function measuring similarity between documents d and e , and $SIM_k(d)$ is the nearest neighborhood of document d , and

$$pol(e) = \begin{cases} 1, & \text{iff } \mathbb{C}(e) = C_P \\ -1, & \text{iff } \mathbb{C}(e) = C_N \end{cases} \quad (11)$$

$$disp(e) = \begin{cases} 1, & \text{iff } \mathbb{C}(e) = C_P \\ \frac{|C_P|}{|C_N|}, & \text{iff } \mathbb{C}(e) = C_N \end{cases} \quad (12)$$

where $\mathbb{C}(e)$ denotes class represented by document e – positive or negative, and $|C_P|$ and $|C_N|$ denotes numbers of examples representing positive and negative class in the training set.

Our second approach, Amplified Similarities Sum (ASS), is similar to the one above. Only difference is that we treat whole training set as the neighborhood, i.e. $k = \infty$. Therefore evaluating function becomes:

$$eval(d) = \sum_{e \in \mathbb{E}} (sim(d, e) \times pol(e) \times disp(e)) \quad (13)$$

where $sim(d, e)$ denotes a function measuring similarity between documents d and e , and \mathbb{E} denotes the training set.

To calculate similarity between documents we use well-known TF-IDF scheme [8] with cosine similarity. In that model documents are represented as vectors. Similarity is calculated as a cosine of the angle between representations of two documents.

$$TF - IDF_{t,e,\mathbb{E}} = TF_{t,e} \times IDF_{t,\mathbb{E}} \quad (14)$$

where

$$IDF_{t,\mathbb{E}} = \log \left(\frac{|\mathbb{E}|}{|\{e: e \in \mathbb{E} \wedge t \in e\}|} \right) \quad (15)$$

$TF_{t,e}$ is a frequency of the term t in the example e from the training set \mathbb{E} , and $IDF_{t,\mathbb{E}}$ is a measure of rarity of the term across whole training set.

Since IDF measure prefers rarely occurring terms, it may not be best suited for Opinion Mining algorithms, e.g. in corpus containing opinions of the users concerning phones they are using, brand name like *Samsung*, *Nokia* or *HTC* would occur less often, than opinion specific words like *good* or *ugly*. So brand name would have higher IDF value, thus it would become more important. Therefore we propose replacement of IDF value with absolute value of term evaluation consistent with score or proportional method [5].

6 Experiments

6.1 Test Set

The main objective of experiments was to test the accuracy of the classification algorithm proposed in Section 5. We used collections of opinions harvested from the site *Znany lekarz*, which gathers opinions about physicians. Each opinion is linked a grade on a scale from 1 to 6. We have assumed that opinions associated with grades 1 and 2 are negative, and opinions with grades 5 and 6 indicate a positive feedback. The dataset contains 2380 negative opinions and 11 764 positive opinions.

Additionally we created second data set, containing equal number of positive and negative documents, by removal of 9384 randomly chosen positive opinions from the collection described above. The resulting set contained 2380 positive and 2380 negative opinions.

In further of this paper we refer to first dataset as *lekarz* and to second one as *lekarz_eq*.

6.2 Performance Measures

To evaluate the effectiveness of the classification we have used two measures. First one is well-known classification accuracy (A) described by equation:

$$A = \frac{t_p + t_n}{t_p + f_p + t_n + f_n} \quad (16)$$

where t_p and f_p are true positives and false positives (i.e., numbers of positive examples from the test set classified correctly and incorrectly), and t_n and f_n are true negatives and false negatives (i.e., numbers of negative examples from the test set classified correctly and incorrectly). In addition to it we have used binary classification quality (Q) [5], which is similar to the F1 measure, and takes into account precision and recall achieved in both classes. This measure is expressed by equations:

$$Q = \begin{cases} \frac{4}{\frac{1}{rec_P} + \frac{1}{rec_N} + \frac{1}{prec_P} + \frac{1}{prec_N}}, & \text{iff } 0 \notin \{rec_P, rec_N, prec_P, prec_N\} \\ 0 & \text{iff } 0 \in \{rec_P, rec_N, prec_P, prec_N\} \end{cases} \quad (17)$$

where

$$prec_P = \frac{t_p}{t_p + f_p} \quad prec_N = \frac{t_n}{t_n + f_n} \quad (18, 19)$$

$$rec_P = \frac{t_p}{t_p + f_n} \quad rec_N = \frac{t_n}{t_n + f_p} \quad (20, 21)$$

where t_p, f_p, t_n, f_n are the same values as in the definition of classification accuracy.

6.3 Experiment Setup

We have performed the 10-fold cross-validation experiments with document representations based on terms and n-grams. In our experiments on classification using term representation we have performed tests using lemmatization, with *morfologik-stemming* [12], stemming, with *Stempel* [1], and without use any text preprocessing method. We also have tested impact of removal of stop-words and rarely occurring words. We have derived the stop-word list from Polish *Wikipedia* [15]. As rarely occurring words we treated those that appeared in fewer documents than β :

$$\beta = \left\lfloor \frac{|C_*|}{|C_{\#}|} \right\rfloor + 2 \quad (22)$$

where C_* denotes the majority class and $C_{\#}$ denotes the minority class in the training set. In experiments on n-gram representation we have tested impact of replacement of IDF value, with a value assigned to a token by proportional or score method, on classification performance. We have also run experiments in which we

removed tokens occurring in fewer documents than β threshold. All experiments were performed on texts converted to lower case. As size of neighborhood k we have assumed 11 and as length of n-grams we have assumed 5.

7 Results

In this Section we present the results obtained by running all combinations of test described in Section 6. The results show that replacement of IDF value with values assigned by score or proportional method tends to increase the quality and accuracy of classification. In most cases the removal of rare tokens leads to slight improvement of classification efficiency.

In all Table 1 and Table 2, Q and A are considered better when they have lower value.

7.1 Experiments on Term Representation

Table 1 shows that performance of classification is worsened when stemming or lemmatization is performed. We see the cause of this behavior in properties of the Polish language, which has a rich grammar [5]. Removal of stop words has minimal and ambiguous influence on classification performance. Removal of rarely occurring terms slightly increases values of classification and accuracy of classification.

Table 1 Sums of ranks from descending rankings of algorithm configurations based on terms, calculated independently for each set based on values of quality Q and accuracy A. Value in each row is the sum of all ranking positions assigned to algorithm configurations. β denotes removal of tokens occurring in less documents than β .

Configuration element value	Q	A
ASS	3456	3699
kNN	2481	1861
NSS	2253	2630
No pre-processing	2234	2317
Stemming	2588	2580
Lemmatization	3368	3293
IDF	2430	2361
Proportional method	1253	1215
Prop. method + β	1422	1495
Score method	1508	1489
Score method + β	1577	1630
w/o stop-words	4658	4523
with stop-words	3532	3667

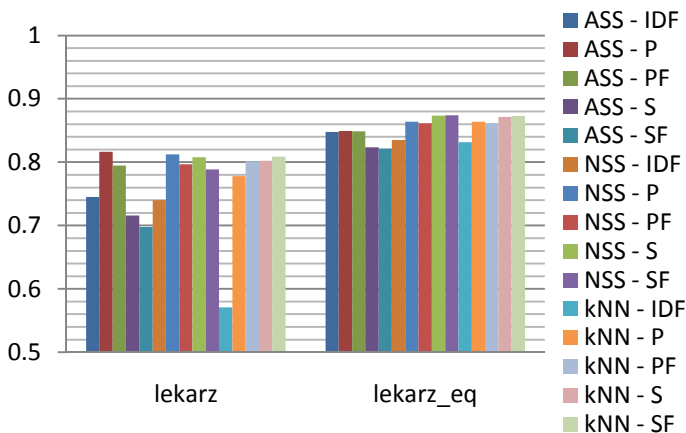


Fig. 1 Average values of classification quality, achieved during the tests using a document representation based on terms. P - proportional method, S - score method, PF, SF - proportional and score method respectively with removal of terms occurring in fewer documents than β

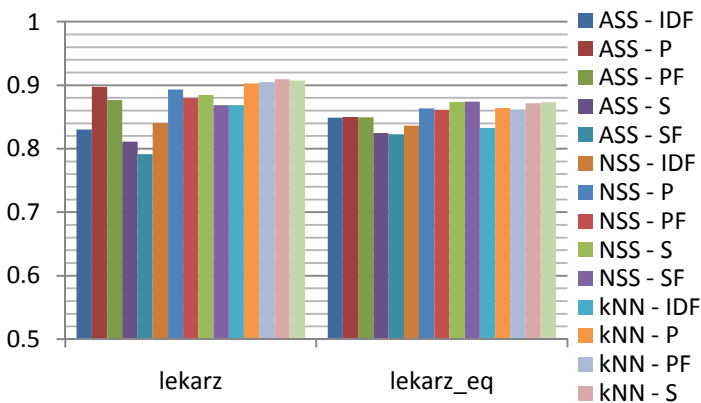


Fig. 2 Average values of classification accuracy, achieved during the tests using a document representation based on terms

7.2 Experiments on n-Gram Representation

As in the case of representation based on terms, removal of rarely occurring terms slightly increases values of classification and accuracy of classification.

Experiments performed using both, term and n-gram, document representations show better classification performance when IDF value was replaced by values assigned to a token by score of proportional methods. NSS gives better

results than classic *k*NN according to quality of classification measure and worse according to accuracy measure. This difference is due to the susceptibility of classical *k*NN methods for disproportion between the sizes of the positive and negative class in the training set.

Table 2 Sums of ranks from descending rankings of algorithm configurations based on n-grams, calculated independently for each set based on values of quality Q and accuracy A. Value in each row is the sum of all ranking positions assigned to algorithm configurations. β denotes removal of tokens occurring in less documents than β .

Configuration element value	Q	A
ASS	101	109
kNN	71	57
NSS	68	74
IDF	71	69
Proportional method	42	39
Prop. method +	48	45
Score method	39	43
Score method +	40	44

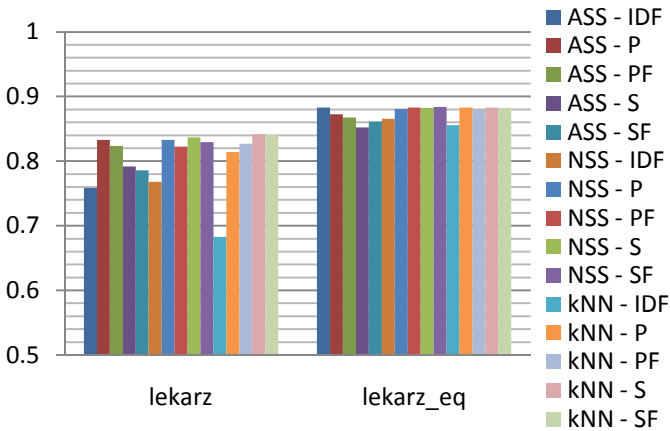


Fig. 3 Average values of classification quality, achieved during the tests using a document representation based on n-grams

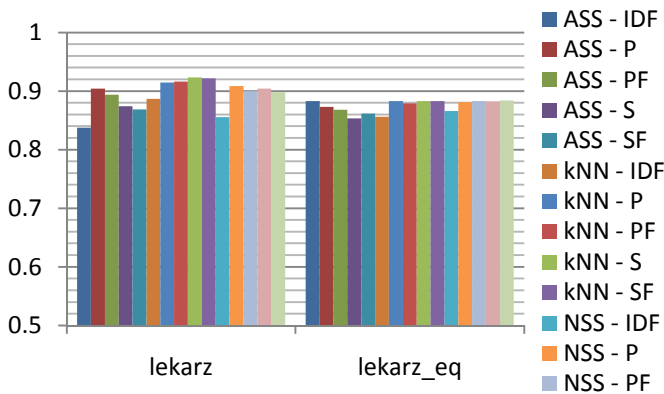


Fig. 4 Average values of classification accuracy, achieved during the tests using a document representation based on n-grams

8 Conclusions

To estimate the performance of *k*NN algorithm there is a need to compare it to method described in [5]. Results are shown in Table 3. Best Q and Best A are the best individual results for combination of other algorithm configuration elements giving best results, not shown in table. Higher values are considered better.

Table 3 Best individual results in experiments in Section 7 and [5]

Data collection	Representation type	Algorithm	Best Q	Best A
lekarz	term	Score	0,8195	0,8917
lekarz_eq	term	Score	0,8892	0,8893
lekarz	n-gram	Score	0,8817	0,9334
lekarz_eq	n-gram	Score	0,9124	0,9124
lekarz	term	kNN	0,8389	0,9198
lekarz_eq	term	kNN	0,8905	0,8905
lekarz	n-gram	kNN	0,8418	0,9231
lekarz_eq	n-gram	kNN	0,8837	0,8838

Table 3 indicates that both methods – *k*NN and scoring [5] are giving similar results. However, *k*NN has greater consumption of computing resources – CPU processing time and memory usage, therefore scoring algorithm can be considered a better one.

References

1. Bialecki, A.: Stempel - Algorithmic Stemmer for Polish Language (2004), <http://www.getopt.org/stempel/> (accessed February 12, 2010)
2. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th International Conference on World Wide Web, New York, USA, pp. 519–528 (2003)
3. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European, New Brunswick, Canada, pp. 174–181 (1997)
4. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of the 19th National Conference on Artificial Intelligence, pp. 755–760 (2004)
5. Jędrzejewski, K., Morzy, M.: Opinion Mining and Social Networks: a Promising Match. In: First Workshop on Social Network Analysis in Applications, SNAA 2011, Kaohsiung, Taiwan (2011)
6. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Now Publishers Inc. (2008)
7. Popescu, A.M., Entzoni, O.: Extracting Product Features and Opinions from Reviews. In: Kao, A., Poteet, S.R. (eds.) Natural Language Processing and Text Mining, pp. 9–28. Springer, London (2007)
8. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Technical Report, New York, USA (1974)
9. Turney, P.D., Littman, M.L.: Unsupervised learning of semantic orientation from a Hundred-Billion-word corpus (2002)
10. Wang, G., Araki, K.: Modifying SO-PMI for Japanese Weblog Opinion Mining by Using a Balancing Factor and Detecting Neutral Expressions. In: Proceedings of NAACL HLT 2007, Companion Volume, New York, USA, pp. 189–192 (2007)
11. Wang, G., Araki, K.: A Graphic Reputation Analysis System for Mining Japanese Weblog Based on both Unstructured and Structured Information. In: 22nd International Conference on Advanced Information Networking and Applications - Workshops, AINAW 2008, Okinawa, Japan, pp. 1240–1245 (2008)
12. Weiss, D., Miłkowski, M.: Morfologik-stemming, Morfologik (2010), <http://morfologik.blogspot.com/> (accessed February 12, 2010)
13. Xu, R.F., Wong, K.F., Xia, Y.Q.: Coarse-Fine Opinion Mining – WIA in NTCIR-7 MOAT Task. In: Proceedings of NTCIR-7 Workshop, Japan (2008)
14. Zhang, W., Yu, C., Meng, W.: Opinion retrieval from blogs. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, pp. 831–840 (2007)
15. Stop listy. Wikipedia, wolna encyklopedia (2010), http://pl.wikipedia.org/wiki/Stop_listy (accessed February 12, 2010)

Designing a Software Transactional Memory for Peer-to-Peer Systems

Aurel Paulovič and Pavol Návrat

Abstract. Transactional memory is a rather novel approach to concurrency control in parallel computing, that has just recently found its way into distributed systems. However, the research concentrates mainly on single processor solutions or cluster environment. In this paper we argue, that peer-to-peer systems would require a different design of transactional memory and we present a few of our design ideas, that as we think could be important to a successful implementation of a scalable and resilient transactional memory.

1 Introduction

With the rise of parallel and distributed applications, concurrency becomes developer's daily bread and butter. Traditionally, the concurrency problem has been solved by using explicit locking and critical sections. While this might be straightforward in smaller applications running on a single computer, it quickly becomes complicated and error-prone as the systems grow larger and more complex and the programmers have to reason about hard-to-debug problems like deadlock, livelock, priority inversion or lock convoying.

When diving into distributed systems, explicit locking gets even harder. Managing and tracking locks in the presence of node failures, latency and network splits often requires carefully crafted locking solutions using dedicated locking services and policies. In addition, locking highly contended resources across a large distributed system can easily become a bottleneck due to the introduced blocking.

Aurel Paulovič · Pavol Návrat

Slovak University of Technology, Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia

e-mail: [paulovic,navrat}@fiit.stuba.sk](mailto:{paulovic,navrat}@fiit.stuba.sk)

Peer-to-peer (P2P) systems, as a class of distributed systems, try to provide scalability and effective sharing of computational resources. In contrast to many systems developed for cluster computing, they tend to be decentralized, geographically dispersed and inherently fault tolerant. However, many of P2P's desired features become issues, when trying to concurrently manage and work with shared data. High node volatility (joining and leaving the network), latency, asymmetric connection speeds and network partitions, that are all much more frequent and severe than in typical cluster systems running in a data center, make concurrency control using explicit locks complicated.

Researchers try to alleviate these problems with new concurrency mechanisms and models. One of such novel approaches is transactional memory (TM). So far, TM research has mainly focused on parallelism on a single chip. In recent years attention has slightly shifted towards TM for multiprocessors and cluster computing. However, to our knowledge only a little work has been done on the topic of TM for P2P systems.

In the rest of this paper we first briefly introduce TM as a concurrency control mechanism. We follow with a description of distributed TM and the differences between TM for cluster computing and P2P systems. Finally, we present our preliminary ideas and design proposals on a software transactional memory for peer-to-peer systems.

2 Transactional Memory

Transactional memory is a relatively new approach to managing concurrency in parallel and distributed systems, that was first practically demonstrated by Herlihy and Moss[4] as an extension to multiprocessor cache-coherence protocol, and later in software implementation by Shavit and Touitou[11]. TM uses the notion of transactions as a concurrency primitive for memory operations and is often implemented as a form of optimistic concurrency control. Generally, a transaction in TM is a finite sequence of operations that satisfies the failure atomicity and isolation properties, and allows the operations of different transactions to be executed concurrently in isolation. It either completely successfully commits, if no conflict occurred, or automatically aborts and rolls back all its changes to the state before the start of the transaction.

The goal of transactions is to free the developer from the need to use explicit locking to denote and protect critical sections, that access shared variables and could be the subject of race conditions. By removing the explicit locks, we can develop a general concurrency management, that is more effective than coarse grained locking, yet does not expose the complications of fine-grained locking and its issues (e.g. deadlock). Also, since transactions can be generally managed at higher levels of concurrency at runtime and not at the level of locks in code, they should provide easier composability of operations that need synchronization.

3 Distributed Transactional Memory

The research of transactional memory for distributed systems has been mainly driven by the rise of cluster computing, data centers and cloud. However, the behaviour of TM on cluster is significantly different from that on a single chip or shared-memory architecture. The latency and communication cost of distributed systems allows the TM to invest more computational time into the concurrency detection and resolution algorithms, which can in turn lower the transaction abort-rate.

Several implementations of distributed TM have been made by the researchers from which some focus on a small number of computing nodes [6, 5], while others were designed to be deployed on larger clusters comprised of hundreds of computing nodes [2, 1]. Also, a distributed TM framework with pluggable support for different transaction synchronization and coherence protocols has been developed [10, 12]. Transactional memory for P2P systems, has been studied rather sparsely [8, 9] or focused on transactional behaviour using locking [7].

4 Differences between Cluster and P2P STM

Peer-to-peer networks typically consist of hundreds or thousands of highly unreliable, geographically dispersed peer nodes, that can almost arbitrarily leave and join the network at any time. The latencies can be one or two orders of magnitude greater than those in a high-end data center and some nodes might have asymmetric connections. Based on that, we have identified a set of differences between cluster and P2P environments, that could prove to be important in the design of a TM.

Nodes in a P2P system are much more likely to fail or to get split from the other nodes by a network partition. TM for P2P networks should probably focus on optimistic concurrency since detecting node failures is relatively complicated and pessimistic concurrency could introduce blocking. Also in the case that a node running a transaction fails, it should not hold any locks or otherwise block the rest of the system.

Since P2P nodes are volatile, data needs to be replicated. This can be true for cluster environment as well, however, there might be a difference between the possibilities of data replication and migration. The peer nodes are often not dedicated solely to the purpose of running the computation or application and could enforce strict limits on the memory used (and would typically have much less memory than a server in a data center) or might be unwilling to host replicated data from other peers because of privacy or performance issues.

Nodes in a P2P network could be potentially malicious. To allow more secure work with data, some resources might be limited to a privileged group of peers that have direct access to them, while other nodes would have to access the resources remotely. TM could support executing transactional operations on remote peers to allow for this scenario.

Distance and latency between nodes in a P2P system is much greater than in a cluster, also the connection speed and throughput are typically considerably lower than between nodes in a data center. The conflict detection and resolution algorithms of a TM should focus on minimizing transaction abort-rate and communication overhead. In addition, a peer network connection might be asymmetric and could prohibit transferring large amounts of data from the node to the rest of the system. TM for P2P should be able to exploit data locality and communication batching.

Because of the nature of P2P systems and their often decentralised architecture, that provides the individual peers only with a partial view of the entire network, locating data items and their replicas is much more complicated. Structured and unstructured networks would potentially require different data search and discovery services and a potential TM implementation should be able to support them.

5 Design Proposals for P2P STM

Since the latency, network partitions and nodes volatility of a P2P system tend to make the communication between peers slower and generally harder, we believe, that lowering the abort rate of transactions and the communication overhead is essential. Therefore, we propose to focus on computationally more expensive conflict resolution and avoidance algorithms, of which cost is diminished by the network I/O, and the relaxation of consistency semantics. In compliance with our observations, we try to draft a few design ideas, that could shape our future work and help us with the implementation of an effective solution for P2P TM.

5.1 *Difference Operations*

Latency, replication and slow network speed makes synchronizing the versions of data items very slow, which in turn increases the probability of data contention. In order to minimize contention and abort-rate, a higher-level conflict detection and resolution mechanism should be used.

One way of achieving this could be the use of *difference operation*(diff), that would not require the TM to write an absolute value to the data item. Instead, they would perform a (possibly commutative) relative operation, that could be applied to the data item later, after the transaction commits without any conflict with similar *diffs*.

An illustration of this idea would be a counter data item with `add` and `subtract` *diffs*. These operations could be executed within a transaction without the need to actually transactionally read the value of the data item first. This would not only save bandwidth and communication, but would also reduce the read set of the transaction and the potential conflicts bound to it. The operation could be sent upon commit to

each replica and be applied there regardless of possible changes made by other such commutative *diffs* executed by concurrent transactions.

If a data item could be changed only by commutative (diff) operations, it would be essentially conflict-free on write. If the operations were to be not commutative, it would imply ordering of the conflicting transactions, that is required by TM. However, the ordering cost in P2P network could be potentially alleviated by the use of eventual consistency.

5.2 *Eventual Consistency*

Following the use of eventual consistency in recent NoSQL data stores, we think that eventually consistent transactions could help with TM scalability. The foundations of eventual consistency for TM have been just recently grounded more formally by Burckhardt *et al.* [3].

The goal of supporting eventual consistency is increased scalability and availability of the system in the presence of frequent network partitions and node volatility. This could for example allow us to build P2P applications that can temporarily disconnect from the network (e.g. a smartphone or a laptop client), continue with work, commit transactions that require to read only local data and synchronize them with the rest of the system, when the client rejoins the network.

Difference operations could reinforce this model even more, because they could allow the disconnected client to perform transactions, that do not use only local copies of data but also remote data via *diffs*. This would also lower the potential of a conflict between the disconnected client and the rest of the system after rejoin.

Eventually consistent memory model would also speed up the commit phase of TM, since it would allow the system to commit changes only to a fraction of the data item replicas and let the changes to be propagated to the rest of the system independently. Such TM would than use quorum to read and potentially update data items with older versions, and preferably *diffs* for writes.

5.3 *Control-Flow*

Control-flow (used in several TMs) is a model in which transactional operations are not performed locally on downloaded copies of data but are sent to the remote node, on which the data item resides. This is the opposite of data-flow model, in which data items are temporarily copied from their respective remote nodes to the node that is running the transaction. While control-flow might lower the total data-transfer in the network, it increases the performance requirements on the remote node, that has to perform the computation.

However, control-flow in P2P system could solve the issues of data security, asymmetric connections and also exploit the raw processing power of selected super-peers. The P2P application could for example use small client peers, that

have limited resources (e.g. battery powered devices) and that would perform only lightweight computation, and server super-peers, that would do the heavy lifting. Other examples would include data with restricted access, like an account, that is located on a secure server, but can be transactionally operated from remote peers through a secure validating interface, etc.

6 Conclusion

Transactional memory as a new concurrency mechanism has been explored in the last two decades mainly in academia and has slowly started orienting on distributed systems. However, it has not been thoroughly studied in the context of P2P systems, which can be very different from the typical cluster environment.

In this paper, we have identified several key differences between cluster and P2P systems, that would have a large impact on the architecture requirements and performance properties of a TM. We have also proposed a few preliminary design ideas and directions, namely difference operations, eventual consistency and control flow, that could enable a successful TM implementation and exploit the nature of P2P networks.

We believe, that using transactional memory could simplify the development of highly available and reliable distributed systems and would abstract the issues of node failures, replication and network partitions from the developer, while still providing good parallelism and concurrency. In this sense, we see TM as an enabling technology that could give birth to a whole new breed of cooperating distributed applications.

Acknowledgements. This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0233-10 and by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11.

References

1. Aguilera, M.K., Merchant, A., Shah, M., et al.: Sinfonia: a new paradigm for building scalable distributed systems. In: Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles, SOSP 2007, pp. 159–174. ACM, New York (2007)
2. Bocchino, R.L., Adve, V.S., Chamberlain, B.L.: Software transactional memory for large scale clusters. In: Proceedings of the 13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP 2008, pp. 247–258. ACM, New York (2008)
3. Burckhardt, S., Leijen, D., Fähndrich, M., Sagiv, M.: Eventually Consistent Transactions. In: Seidl, H. (ed.) ESOP 2012. LNCS, vol. 7211, pp. 67–86. Springer, Heidelberg (2012)
4. Herlihy, M., Moss, J.E.B.: Transactional memory: architectural support for lock-free data structures. SIGARCH Comput. Archit. News 21(2), 289–300 (1993)

5. Kotselidis, C., Ansari, M., Jarvis, K., et al.: DiSTM: A Software Transactional Memory Framework for Clusters. In: Proceedings of the 2008 37th International Conference on Parallel Processing, ICPP 2008, pp. 51–58. IEEE Computer Society, Washington, DC (2008)
6. Manassiev, K., Mihailescu, M., Amza, C.: Exploiting distributed version concurrency in a transactional memory cluster. In: Proceedings of the Eleventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP 2006, pp. 198–208. ACM, New York (2006)
7. Mesaros, V., Collet, R., Glynn, K., et al.: A Transactional System for Structured Overlay Networks (March 2005)
8. Müller, M.-F., Möller, K.-T., Schöttner, M.: Commit Protocols for a Distributed Transactional Memory. In: Proc. of the 2010 International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2010, pp. 1–10. IEEE Computer Society, Washington, DC (2010)
9. Pratt-Szeliga, P., Fawcet, J.: p2pstm: A Peer-to-Peer Software Transactional Memory. Technical report, Syracuse University, Syracuse, NY (2010)
10. Saad, M.M., Ravindran, B.: Supporting STM in Distributed Systems: Mechanisms and a Java Framework. In: TRANSACT 2011: Proceedings of the 6th ACM SIGPLAN Workshop on Transactional Computing, San Jose, California, USA (June 2011)
11. Shavit, N., Touitou, D.: Software transactional memory. In: Proceedings of the Fourteenth Annual ACM Symposium on Principles of Distributed Computing, PODC 1995, pp. 204–213. ACM, New York (1995)
12. Turcu, A., Ravindran, B.: Hyflow2: A High Performance Distributed Transactional Memory Framework in Scala (April 2012)

Traceability in Software Architecture Decisions Based on Notes about Documents

Gilberto Pedraza-Garcia and Dario Correa

Abstract. In general, the results of a software architecture project are a great deal of heterogeneous artifacts in its final version. In order to improve and make a more efficient process of software architecture definition, architects in charge require to understand the ways as it was defined and the justifications for each item and element. During the development of each iteration intermediate versions are documented about artifacts that serve architects to exchange opinions, concerns, directions, possible alternatives, tradeoffs of the architectural advances. Such considerations are a fundamental piece of knowledge for improving the process. Nevertheless, many of those are verbal, they are not documented and therefore they are forgotten with the pass of time. A strategy is proposed in this research work for knowledge management to capture, represent, formalize, code and trace design considerations expressed as notes in documents. By using ontologies, architects apply and reuse knowledge obtained in Notes. This strategy is validated through a software architecture project belonging to the digital printing sector.

Keywords: Software Architecture Knowledge Management, Architecture Traceability, Architecture Documentation Notes.

1 Introduction

This work is part of a project for knowledge management that seeks to improve efficiency, increase success possibilities and assess quality in software architectures with the approach of Viewpoints and Perspectives by applying decision

Gilberto Pedraza-Garcia · Dario Correa
Universidad de Los Andes, Departamento de Sistemas y Computacion
Cra 1E No 19A-40, Bogota, Colombia
e-mail: [g.pedraza56, dcorreal}@uniandes.edu.co](mailto:{g.pedraza56, dcorreal}@uniandes.edu.co)

traceability strategies in software architectural design and Aspect Oriented Engineering principles.

Software architecture defines a high level structure and it is key to guarantee evolution, sustainability and communication in a software system [2]. To that end, it is required to understand how and why a system is built [11]. There are several approaches for study and review software architectures. One of the most used is Viewpoints and Perspectives [8]. Such an approach provides a set of guidelines about aspects that should be reviewed and the ways to do it; however, this approach is focused on explaining the resulting artifacts from the process by mean of views and a few aspects about the architecture development. If products out of the software architecture making process are kept in mind, these are not limited just to a set of architecture models and views [6]. The result is a partial documentation that does not reflect neither how the architecture was made nor the considerations taken into account by the architect in order to defining each of the architectures items. There is also the need of an explicit and detailed description of the decisions made, alternatives taken into consideration, assessments, opinions, suggestions, advantages, disadvantages, justification and corresponding reasoning. As a consequence, the software maintenance costs are increased. There is currently a rising approach that defines software architecture as the result of a set of decisions that justify the architecture design [3] and they do have an affectionation on the system quality.

The explicit documentation about the reasoning applied on architecture design decisions is an accepted practice by architects. However, this is implemented very rarely in industry [4]. This is due to the fact that the general perception is that architects do not understand the critical roles of the systematic use, the reasoning capture and design justification as a whole [9]. In such a way, a good part of knowledge produced in the architecture definition process is implicit, it increases the architect's expertise, but due to the fact that this is not coded (documented), it is not shared or reused and it disappears with the pass of time. Specifically, not having an applied and reviewed reasoning on decisions by the end of the process of making a software architecture makes the tasks of explaining the architecture client or owner the architecture design more complex and expensive, tracing causes whenever there is a defect in software or tracing the design evolution [11].

The main objective of this research work is to develop a knowledge management strategy in order to improving the success and quality possibilities in making software architectures with the approach of viewpoints and perspectives as proposed by Rozanski [8]. Specifically, this work is intended to understand the ways as software architects make design decisions to incorporate those elements into a decision making model. With this model, we look to make risks and costs quantification possible in decision making. It is hoped that the model could capture insight knowledge in order to code it, share it and reuse it in other projects and with other software architects. The strategy is based on application of several instruments such as interviews and surveys to determine the ways as architects make design decisions.

For capturing implicit knowledge there will be a standard template based method incorporated into the software architecture defining process. Such a knowledge is fed on an architecture decision making model represented by an ontology describing related concepts. At the same time, said ontology models the traceability between architecture items and design decisions, which allows to infer new relationships between architecture decisions and items. The models of viewpoints and decisions are intertwined by mean of Aspect Oriented Modeling to keep independence between such models.

This work is organized as follows: Section 2 describes the problems of recording architectural documents. Section 3 introduces a background about the knowledge management and reasoning on architecture design. Section 4 depicts the ontology based solution strategy. Section 5 introduces implementation details and experiments in process. Section 6 presents the related works and Section 7 shows this research work conclusions.

2 Case of Study of Notes in Architecture Documentation

In software architecture definition architects exchange ideas of several kinds whereby they all express concerns about the general architecture or its key particular elements. Generally speaking, such considerations are verbally expressed, shared on e-mails, wrote about in paper documents or they are made by mean of comments in word processors named as notes. In such a way the set of considerations and messages exchanged with and between architects within the architecture definition process is turned into a valuable knowledge in order to understand, explain and verify architecture in absence of responsible architects because it keeps reasoning and / or justifications as applied in each of the software architecture decisions [11].

This research work is focused in chronologically organize considerations made by architects in order to make an easier understanding of the architecture definition process and it allows to establish improvements because these constitute real and true traces about the ways as the architecture items were built: changes in one version to another, restrictions, quality attributes among other aspects that were taken into consideration. This research work proposes a strategy to give structure and capture considerations made by architects in the architecture definition process in which they whether clarify, explain or justify several decisions upon software architecture items. Reasonings are captured from software architecture documents, then taken towards an ontology whereby concepts related to architecture artifacts are unified in order to allowing specific searches upon notes, chronological surf about architecture decisions through several versions of documents, support further decisions whether in the same or in another project, establish new relationships between architecture decisions and items.

3 Background

3.1 Reasoning Based Knowledge Management in Software Architecture

This research work is oriented towards the knowledge management as acquired from reasonings that architects apply in architecture definition, as proposed by Tang et al. [10]. Reasoning knowledge is the support for architecture decisions because it includes the reasons justifying each decision made. The focus as proposed is fit into the Bosch proposal [3], who considers architecture as the "composition of an architecture decisions set." In this way, architecture decisions and their corresponding reasoning are important knowledge elements of architecture in a software system [1]. The software architecture design reasoning allows to know and understand stakeholders interests as considered for each decision, associated restrictions and arguments used to choose a particular one [1]. In the practice, the architecture knowledge is often lost or vaporized resulting in trouble such as maintenance cost increase [3].

3.2 Note Traceability in Software Architecture Items

Note traceability is the ability to describe and monitor the lifecycle of making each of the architectural artifacts by mean of considerations exchanged by responsible architects. Particularly, it allows to go through each of the notes made by architects when defining each element placed in architecture. Moreover, it allows to explain the architecture design, reasoning and decisions upon each of the architecture items. Tang et al., [11] refers to the following contributions of architecture definition traceability: To explain architecture design by traces on reasonings and decisions, to identify impacts in requirement changes and decisions, to identify defect injection points, to verify architecture design, to manage the evolution traces of the system, to relate and group architecture design objects with or to more understandable concepts such as requirements, restrictions and / or assumptions.

3.3 Ontology: Tools for Knowledge Management

Ontologies allow to make models of part of the reality and its interrelations. By mean of ontology reasoning applied knowledge is organized by architects in their architectural decisions. Then, this is taken to a repository where storage structures are normalized; mechanisms are offered to make general and / or specific searches or it is necessary to think to obtain new traces between architecture decisions and items. In such a way, ontology becomes a very important knowledge reuse mechanism. In an ontology there are two big components observed: the scheme and data or concept or individual stages [13].

4 Solution Strategy

Keeping in mind the method proposal by Van Vliet [1] for knowledge management, there are 5 activities which are described as follows:

4.1 Note Model Derivation

In this activity, the main knowledge entities of the process are identified. In this case, these are related to interaction or communication between architects participating in defining architecture. Document supports were reviewed as they belonged to former software architecture projects which were made by a group of architects. With these messages, there is the creation of a set of note templates as described here on:

- **Request for Clarification Notes.** These are clarifications, observations and / or corrections about features described in architecture and these are of interest for a stakeholder or another architect.
- **Opinion Notes.** These are fellow architects contributions with approaches, directions and / or suggestions about future decisions, or these were made before, or corresponding to items described in architecture, or they have an optional character for the architect in charge to make new decisions.
- **Mandatory orientation Notes.** These are requests made by architects in charge with directions or opinions that have a mandatory character with strategies to solve a particular situation.
- **Decision Notes.** These are descriptions about decisions made by the architect team, as observed in figure 1

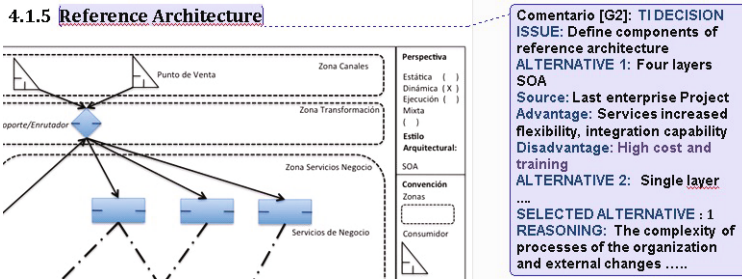


Fig. 1 Decision Note Example

- **Assessment Notes on Quality Attributes.** These are valuations made by grading or assessing architects about the impacts of decisions upon the system’s quality features. By mean of scenarios describing particular situations to assesss, they involve a set of architecture items that are sensible and then tradeoffs are analyzed about such a decision.

- **Disagreement Notes.** These allow to describe possible disagreements between architects responsible for the project about decisions defining an architecture item. By mean of a scenario, one or several architecture decisions are analyzed and assessed at considering risks, sensible elements and tradeoffs, likewise justifications to accept or reject a decision.
- **Open Notes.** These open notes are texts with comments that have no structure but they do have a justification on a decision, directions, valuations, disagreements, suggestions and / or clarifications that an architect could make. In this research work it is proposed to give those an structure as notes and include them into the traceability model. To that end, for each note, an associated decision is created in the document number. Notes in the same number but with former versions are considered solution alternatives, and comments with the last version are defined as decision reasoning. The chosen alternative corresponds to the text, chart, figure or artifact associated to the number.

4.2 Note Capture of Software Architecture Documents

Normally, a document version is generated per iteration in order to support advances reached. In addition, each architecture document is sorted out in sections and it has a certain scheme of hierarchical number, which corresponds logically to the set of architecture items. In this way, a section or set of numbers relate an architectural device to a set of notes or comments left by architects in the corresponding sections of the documents. Notes are obtained by mean of a parser of architecture documents by extracting notes and features associated to the document separately such as follows: author, date, time, section or number appointed.

4.3 Formalization and Integration into the Notes Model

Captured knowledge by note templates is formalized with the help of ontologies. In figure 2 the main ontology concepts are presented:

4.3.1 Note (Anotation)

It is the most abstract concept to express any considerations made by the architect about software architecture. These include features such as subject, restrictions, search key words, current status, role an position of the proposing architect, applied reasoning, one or several references to artifacts or architectural items, date and time. Concrete notes correspond to the ones described in section 4.1 and these are related to the words "is-a-note".

4.3.2 Version

This is the highest hierarchy concept to describe the software architecture structure. For this project an architecture could have several versions. Each of those is developed in stages or phases. Phases make refer to diverse domains, each of those has architectural artifacts and artifacts have items and elements. Notes make "reference" to versions, phases, domains, artifacts or items and elements.

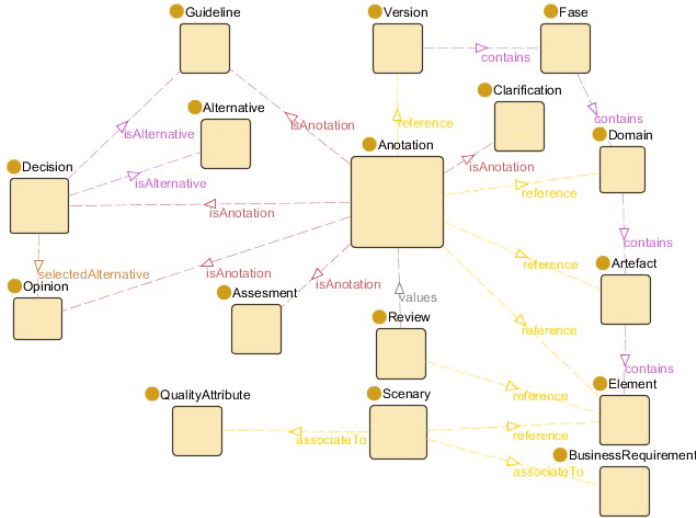


Fig. 2 Main concepts represented in the Note Ontology

4.3.3 Task

This defines common aspects and contents in notes. Scenarios describe particular situations to make assessments or propose disagreements. Scenarios make use of reviews to assess a concrete decision. Reviews identify potential risks, sensitivity points in architecture and analyze tradeoffs. Alternatives are used by decisions to define possible solutions (is Alternative) and one of those is the chosen one (selectedAlternative). Alternatives have a description (description), a source, advantages and disadvantages (tradeoff). Solution alternatives, guidelines or opinions are also alternatives (isAlternative) for decision making.

4.3.4 Requirement

It expresses stakeholders interests about software architecture. These could be business requirements or quality attributes. Requirements are associated to scenarios.

4.4 Note Repository Updating

Each note extracted from documents is broken into concepts expressed in the former number and some features are added such as author, document reference, time and date. When adding a new note to the model, this represents it as a stage or individual in ontology, to which an only identifier, relations towards other individuals and values of its properties are given.

4.5 Knowledge Applications and Reuse

After having an explicit knowledge obtained from notes and their interrelations, the former is ready to be used by stakeholders in the project by mean of the following mechanisms: First, surfing upon associated concepts and stages in ontologies at the levels of version or between versions of documents. In this way, the chronology of an item or set of items could be rebuilt in architecture. Second, specific searches on ontology by making reference to a particular item to satisfy specific interests by a Stakeholder. Third, in a new project, an architect is able to select, reuse and filter notes from former projects online. Fourth, traceability of evolution in each architectural item by mean of traces between available versions of the document.

5 Implementation, Experiments and Validation

The proposed strategy applies to the field of technology of a software architecture project for an organization in the digital printing sector. This company seeks to make their business strategies more dynamic by the intensive use of state-of-the-art technology. Architecture is developed increasingly with an iteration based method by following recommendations and proposed activities by the TOGAF version 9 framework, with a time width of 36 months to obtain the structure as expected. In this project there are people participating on behalf of the company and a software architectural team made up by a project manager and two architects. For the implementation of the solution the following tools are being built: First, an Add - in for MS-Word processor that allows to add notes as described in section [4.1](#). This could be done in form of comment templates associated to the text, as observed in figure [1](#). Second, a software for extraction and capture of notes from the architectural documents by mean of the Interop.Office framework by Microsoft.Net. Third, an ontology designed with the Protege Editor, whereby concepts issued in the described model were defined in section [4.3](#). To add stages to this ontology, a .Net application is built by using the JENA framework for ontology management. This provides a set of functions to develop web - semantics based applications. This is done by mean of using RDF technologies. Fourth, a Note repository that is a MySQL database. Fifth, an inference mechanism to make explicit new traces on ontology by using the

PELLET library, a semantic thinker that allows to make inferences and issue new relationships between ontology individuals from a set of facts and axioms.

Experiments are made by helping and advising the software architecture development for the aforementioned digital printing company in five (5) iterations with their corresponding versions of the architectural documents. In the first place, analysis and classification were made on the total set of found notes. The purposes of said notes were sorted out as clarification, opinion, direction, decision, quality attribute assessment and disagreement. To achieve this, the documents from two (2) final architectural projects by the architect team were used. These contained notes without any structures in forms of MS-Word comments; others were made in written paper and another group was found in e-mails exchanged between architects in charge. From all those documents the notes model was defined as proposed in the section [4.1](#).

Notes from the architectural documents are being currently collected and processed simultaneously with the development of iterations by architects. There is a note database in ontology and searches have been achieved upon architectural items from previous versions, hierarchical surfing of notes and generation of basic traces between notes and architectural elements and items. Although the process is not completed yet, interesting data have been obtained to adjust the model proposed for notes. For example, it was found that suggestion, opinion and direction notes were used indistinctively, this is, architects found no difference between all those three kinds. Moreover, architects complained of high complexity in assessment type notes. Adjustments in the notes model shall be made in the next version.

6 Related Work

Kruchten, Lago and Van Vliet [\[5\]](#) propose a case model of use for a knowledge database of architecture supported on an ontology describing decisions on architectural design. Kruchten defines a typology in terms of existent decisions that are materialized into visible architectural items. Such decisions could be structural or behavioral. In this research work a set of scenarios is proposed to associate architectural decisions to quality attributes. Executive decisions as proposed by Kruchten correspond to decisions that involve entrepreneurial architecture domains that are different to software architecture. Tyree and Akerman make a standard template-based proposal to give structure to architecture decisions by associating concepts such as problems (issues) that have orienting principles (assumptions). Each decision has a set of feasible options or alternatives from which the best is chosen according to a reasoning (argument). The chosen alternative has implications and affects other decisions (related decisions) [\[12\]](#). Liang et al.,[\[7\]](#) propose Knowledge Architect a suit having several tools: Knowledge repository, which provides several interfaces for storing and retrieving architectural knowledge; Document Knowledge Client, which is a plug-in word that allows capturing information by notes in requirement and architectural documents. Finally Knowledge Translator, a

semi-automatic tool that translates software architectural knowledge expressed in a domain model to a domain where stakeholders would be able to understand architecture.

7 Conclusions

A knowledge management strategy is presented in this research work on software architectures in order to make an easier understanding and improve the definition processes of architecture. The strategy is based on capturing and processing considerations exchanged by architects during the making processes of several versions of architectural documents. In such a way, the responsible team for software architecture can review and assess the developed process and reuse that corpus of knowledge that is relevant for further projects. One of the contributions of this research work is a model for understanding the architecture definition process based on traceability of notes in the architecture documents. It also offers alternatives to code and make explicit knowledge communicated by architects in their definition processes. Moreover, it shows the path to develop a software architectural approach whereby decisions are turned into first class entities. As a future work there is the incorporation of the model and its results to support architecture maintenance and evolution by focus and direct it towards traceability in design decisions.

References

1. Babar, M.A., Dingsoyr, T., Lago, P., van Vliet, H.: Software architecture knowledge management: theory and practice, ch. 2, 3 and 8, pp. 21–38, 39–57, 137–154. Springer (2009)
2. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley, Boston (2003)
3. Dannenberg, R.B.: Software Architecture: The Next Step. In: Oquendo, F., Warboys, B.C., Morrison, R. (eds.) EWSA 2004. LNCS, vol. 3047, pp. 194–199. Springer, Heidelberg (2004)
4. Falessi, D., Capilla, R., Cantone, G.: A value-based approach for documenting design decisions rationale: A replicated experiment. In: SHARK 2008 (May 2008)
5. Kruchten, P., Lago, P., van Vliet, H.: Building Up and Reasoning About Architectural Knowledge. In: Hofmeister, C., Crnković, I., Reussner, R. (eds.) QoSA 2006. LNCS, vol. 4214, pp. 43–58. Springer, Heidelberg (2006)
6. Peng, L., Jansen, A., Avgeriou, P., Tang, A., Xu, L.: Advanced quality prediction model for software architectural knowledge sharing. *The Journal of Systems and Software* 84, 786–802 (2011)
7. Peng, L., Jansen, A., Avgeriou, P.: Knowledge architect: a tool suite for managing software architecture knowledge. Tech. rep., University of Groningen (February 2009)
8. Rozanski, N., Woods, E.: Software systems architecture: working with stakeholders using viewpoints and perspectives, 2nd edn. Addison-Wesley Educational Publishers (2012)
9. Tang, A., Ali Babar, M., Gorton, I., Han, J.: A survey of architecture design rationale. *The Journal of Systems and Software* 79, 1792–1804 (2006)

10. Tang, A., Avgeriou, P., Jansen, A., Capilla, R., Babar, M.A.: A comparative study of architecture knowledge management tools. *The Journal of Systems and Software* 83 (2010)
11. Tang, A., Yin, Y., Han, J.: A rationale-based architecture model for design traceability and reasoning. *The Journal of Systems and Software* 80, 918–934 (2007)
12. Tyree, J., Akerman, A.: Architecture decisions: demystifying architecture. *IEEE Software* p. 19 (March-April 2005)
13. Wang, C., Lu, J., Zhang, G.: Integration of ontology data through learning instance matching. In: *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 536–539 (December 2006)

OLAP Models for Sequential Data – Current State of Research and Open Problems

Łukasz Nienartowicz

Abstract. In recent years, sequential data processing has been extensively studied in the research literature. Because of the popularity and peculiarity of this type of data, many systems devoted to storage, sharing and processing of sequential data have been created. The development of this type of systems includes databases with SQL-like languages, data warehouses and OLAP. So far, four models of OLAP cubes for sequential data have been designed: *FlowCube*, *S-OLAP*, *OLAP on Search Logs*, and *E-Cube*. These models significantly differ from each other. In effect, there is a need to analyze these models and compare their usability. The following analysis reveals advantages and disadvantages of the aforementioned models and discusses possible research issues.

1 Introduction

Sequential data are omnipresent. Every day people carry out a variety of activities to achieve their desired effect. For example, to travel from one point of a city to another, we must usually use a few means of transportation. In factories, the final product is composed of semi-finished products in an appropriate order. Large amounts of sequential data are also produced by: radio-frequency identification (RFID) systems, customers who are using online stores, intelligent building management systems, and applications which write logs.

The popularity of sequential data has decided about the importance of their analysis. Sequential data are very interesting because they create opportunities to find data patterns or trends. For example, online store owners are interested in how their customers move in the store. They want to explore patterns of behavior of customers who have made a purchase or not. Another example is stockbrokers. They are looking for models which will allow them to predict further behavior of stock market

Łukasz Nienartowicz

Poznan University of Technology, Poznan, Poland

e-mail: lukasz.nienartowicz@doctorate.put.poznan.pl

indices. Even in large companies, such as banks, analysis of data collected in the workflow system promotes continuous improvement of business processes. Also, it is crucial in modern science to search for patterns in strings of data, for example in the DNA code.

The analysis of sequential data is more complex than working with data stored in a relational database. Traditional databases are mainly subjected to analysis of records values, while in sequential data, the information is hidden also in the order of records. It should be noted that events which form a sequence could be treated in two ways: we can either consider the event being active for a period of time or we can treat it discretely as a point in time. The difference between these two approaches was thoroughly described in [23].

There are numerous data mining techniques to analyze sequential data systematized in [15]. Data mining is a large and separate area of science. However, this article focuses on problems associated with sequential data processing, so data mining problems have been omitted.

The universality of sequential data has become a challenge for the IT fields associated with storing and sharing data. The usual relational data structure has not always been sufficient. Therefore, new data structures have been sought for years to help quickly and efficiently find patterns in large collections of sequential data.

Sequential data processing is still a new field of study. To the best of my knowledge, tools that have been proposed previously are still imperfect. The previous research of OLAP cubes for sequential data left space to the development of the already existing solutions or to the creation of new models.

This article has two main aims, the first of which is a review of the research literature focusing on the subject of storage, sharing and processing of sequential data, which is described in Section 2. The analysis includes databases and data warehouses, query languages, and OLAP cubes in particular. The most important part of this section is the comparison of feasibility and efficiency of different models of OLAP cubes for sequential data. The other aim, described in Section 3, is to identify the problems which appeared in the development of OLAP systems for sequential data. Those problems could become potential research issues.

2 Literature Review

The need for sequential data analysis promoted the creation of models of databases which are able to store and share sequential data efficiently. Also, query languages had to be improved to support new types of data manipulation. The first database system which has a formal support for sequential data was PREDATOR described in [19, 22]. PREDATOR introduced a model of the sequential data record. For the purpose of this system, the extended SQL-like query language SEQUIN ([20, 21]) was developed. SEQUIN includes operators, like *Offset* or *Aggregate*, which are required to work with sequential data.

Another interesting language is *Sorted Relational Query Language* (SRQL) described in [16]. SRQL was implemented on DEVise system ([13]), which has many other applications. SRQL is based on SEQUIN but its syntax was extended by new operators. The most interesting of these operators is SEQUENCE BY, which allows to split data into clusters. In each cluster, events represent exactly one sequence.

The extension of SRQL is yet another language - *Simple Query Language for Time Series* (SQL-TS) proposed in [17, 18, 9]. In SQL-TS, the clause SEQUENCE BY is supported by the clause AS, which allows to define a desired pattern. A new operator - CLUSTER BY - specifies a data category which should be processed separately. The language changes when combined with the use of optimization algorithms, which results in high efficiency of searching data patterns.

RFID applications produce such a large number of sequential data that it was necessary to design a RFID Data Warehouse. The conception of such a warehouse from the high-level perspective was introduced in [1]. In [8], the construction and fundamental algorithms for RFID warehouse were produced. This system was supplemented by OLAP-style analysis operations like *roll-up* and *drill-down*.

Naturally, the next step in the development of systems based on sequential data was to create the real OLAP cube. Traditional OLAP systems are powerful tools because they allow to analyze data at different summarization levels. Unfortunately, the traditional OLAP cube model does not allow to analyze data taken as a sequence. Therefore, it was necessary to create a new OLAP model which would work with sequential data at different abstraction levels. So far, four models of OLAP cubes for sequential data have been designed.

2.1 FlowCube

FlowCube described in [6] and extended in [7] supports the analysis of items flow in RFID applications. The data needed to create the cube are stored in Path Database. The data structure is simple. Each record contains: Electronic Product Code associated with a particular item, values describing the product (which are path independent dimensions) and the path of the item composed of the identifier of locations and time duration spent in a given location.

This kind of cube differs from the traditional OLAP because the measure of each cell is not a scalar aggregate but a commodity *flowgraph*. The *flowgraph* contains movement trends and deviations of the item. The graph construction is a tree where each node represents a location and each edge indicates a transition between associated locations. Each edge has an assigned value representing the probability of transition. This probability is calculated as a percentage of items transported by this route.

FlowCube is constructed of a collection of cuboids. A cuboid groups and aggregates items which have the same values at the same abstraction level in one cell. The paths of the items must also be aggregated to the expected detail level. Therefore,

each cuboid can be characterized by a pair of values: the item abstraction level and the path abstraction level.

2.2 *S-OLAP*

The second model, *S-OLAP*, proposed in [2] and extended in [14], was created to analyze the subway passenger flow. This model focused on searching event patterns, which makes it very versatile. *S-OLAP* implements six operations for pattern manipulation: *append*, *prepend*, *de-tail*, *de-head*, *pattern-roll-up* and *pattern-drill-down*. The first two operations add a symbol at the end or the beginning of a pattern respectively. On the other hand, the *de-tail* operation removes the last event from the pattern and the *de-head* removes the first one. The last two operations modify the abstraction level of the pattern dimension. For example, the *pattern-roll-up* may roll up time dimension from day to week, and the *pattern-drill-down* works the reverse.

S-OLAP is composed of sequence cuboids which present answers to the *Pattern-Based Aggregate* query (described in [14, 4, 5]). The language of the query is inspired by SQL-TS and contains five specific parts. Three of them - CLUSTERED BY, SEQUENCE BY and AS - have already been described. The fourth one, i.e. SEQUENCE GROUP BY, allows to define the abstraction level of every dimension in hierarchy. The last clause is CUBOID BY and it defines the format of the sequential patterns to be matched, the behavior if a single sequence contains multiple occurrences of a pattern, and requirements for other event values.

So far, two algorithms of the *S-cuboid* construction have been created. The *Counter-Based* approach is simple and is based on storing a counter in each cell of the cuboid. The counters are incremented when the data sequences are scanned. This algorithm can be efficient only if the number of *S-cuboid* cells is low and can be stored in memory. The other algorithm is based on the *Inverted Index* method, which was first used in an OLAP construction in [10] and used a list of inverted indices. Every Inverted Index is a list of identifiers of all sequences that go with the indicated pattern. The *S-cuboid* is computed by joining suitable inverted indices. In this algorithm, all six operations for pattern manipulation were implemented. The *Inverted Index* approach is much more efficient.

It should be noted that all the algorithms mentioned above were implemented in the system described in [3].

2.3 *OLAP on Search Logs*

OLAP on Search Logs presented in [25] is used to help Search Engines recognize the user's needs and preferences. Search logs are specific history of the user's behaviors while they are searching information. These data can be very useful to help applications like query suggestion or keyword bidding. For example, a review of a search log can help personalize websearch results to facilitate the user's expectations.

It should be noted that major Web Search Engines logs contain billions of searches. Such a huge amount of data is very difficult to be analyzed; thus, *OLAP on Search Logs* must ensure high efficiency.

This kind of cube is based on the three basic search engines mining functions: *forward search*, *backward search* and *session retrieval*. The *forward search* finds top-k most frequent sequences that have an indicated subsequence as a prefix. It is very useful to search and suggest queries that may be interesting for the user. Query suggestion applications are based on this algorithm. Symmetrically, there is a *backward search* operation, which finds the most popular sequence with a typed subsequence as a suffix. This operation is used in keyword bidding applications, which are an important service in a sponsored search. Both operations are used to find a subsequence of a single search. *Session retrieval* may be useful to analyze the user's whole session as a pattern. This function can be used to monitor the search quality and to diagnose why the results do not satisfy users.

The construction of *OLAP on Search Logs* is based on suffix trees to serve a forward search. The suffix tree organizes all existing suffixes into a prefix sharing tree. The structure of the tree is as follows: each edge is labeled by the next element of a sequence and each node except the root, and is associated with a value, which represents the frequency occurrence of the query in search logs. To realize the backward search, the reversed suffix trees were built. It is crucial that the suffix tree can be scaled up to a distributed environment, which helps to scan a search log on the fly.

2.4 E-Cube

The last model, *E-Cube*, proposed in [12, 11], is a combination of OLAP and *Complex Event Processing* (CEP). CEP systems are designed for pattern matching in real time by processing data streams [24]. The basic operator in the query language for CEP systems is SEQ. This operator specifies the order in which indicated event types must occur. The SEQ query language enables to group events by means of attributes and to compute aggregate functions like COUNT. The clause WITHIN is particularly interesting as it specifies the time difference between the first and the last event.

The *E-Cube* model is a directed acyclic graph. In this graph, each node stores a pattern query, while each edge corresponds to the relationship between pattern queries in associated nodes. There are two types of dependencies between nodes, namely *concept* and *pattern*. If there is a *concept* relationship between two nodes, it means that the correspondent patterns are at different abstraction levels. The *pattern* dependence is based on deleting one or more symbols from a query pattern. To find an optimal execution order for queries in an *E-cube* hierarchy, the optimization problem is mapped into a well-known graph.

An individual pattern query and its result instances is called *E-cuboid*. Naturally, *E-Cube* is an integrated collection of smaller *E-cuboids*. There are four operations on *E-Cube*, which are an extension of standard OLAP operations and allow

users to move from one *E-cuboid* to another. The *pattern-drill-down* operation adds one or more events, which are at the same abstraction level, to a query pattern. *Concept-drill-down* transforms one or more events to a higher detail level. Obviously, *pattern-roll-up* and *concept-roll-up* are opposite operations.

2.5 Comparison of Models

The summary of the aforementioned approaches to managing sequential data are shown in Table 1.

Table 1 Comparison of OLAP models for sequential data

	FlowCube	S-OLAP	OLAP on SL	E-Cube
Data type	RFID	universal	Search Logs	universal
Structure	Flowgraph (tree)	Inverted Index (collection)	suffix tree	acyclic graph
Operations on cube		Append Prepend De-head De-tail P-roll-up P-drill-down	Forward search Backward search Session retrieval	Pattern-drill-down Concept-drill-down Pattern-roll-up Concept-roll-up
Incremental update	No	No	Yes	Yes
Data privacy	No	No	No	No
Data integration	No	No	No	No
Events type	interval	discrete	discrete	discrete

All of the models differ from each other significantly. In each case, the method of the cuboid construction is different. Potential applications of the above solutions also vary, hence a need to analyze these models and compare their usability.

3 Research Issues

1. The analysis of the existing models of OLAP cubes for sequential data exhibits many potential research issues areas, the first of which is **performance**.

Designing OLAP on sequential data is considerably more complicated than constructing a traditional OLAP. In a cube on relational data, full materialization is often used. In OLAP on sequential data, full materialization is not available because the number of dimensions is unbounded. Moreover, in the case of data patterns search systems like *S-OLAP*, there is also a problem that the cube is non-summarizable, which means that the cuboid at the higher abstraction level cannot be created from a set of cuboids at a lower abstraction level. This problem is described in [14].

2. In many real applications, the data are continuously generated, which causes two problems. First, some systems may need access to as latest data as possible. Second, when a data cube is extra-large, it is not effective to calculate it again. Because of that, **incremental update** of OLAP systems became a very interesting issue, especially difficult in the *S-OLAP* case, because other models are constructed as a graph, which makes updates much easier.
3. Table 1 clearly indicates that the problem with sequential **data integration** from many sources to one OLAP system is still open. None of the presented models contains components which can be helpful when for example two databases need to be joined to enable more detailed data analysis.
4. Working with data from more than one source generates problems related to **data privacy**. This also occurs in complex organizational structures. The aim is how to preserve confidential information in individual data cells and provide a precise summation of values.
5. The analysis showed that only one model, *FlowCube*, supports events treated as an interval. However, because this model is dedicated only for RFID data, there is a need to design and implement a new universal model for processing **interval sequential data**. Finally, there are not any defined operators which promote effective work with this kind of data.

References

1. Chawathe, S.S., Krishnamurthy, V., Ramachandran, S., Sarma, S.: Managing rfid data. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, vol. 30, pp. 1189–1195 (2004)
2. Chui, C.K.: The design and implementation of an olap system for sequence data analysis. In: Proceedings of the 2nd SIGMOD PhD Workshop on Innovative Database Research, IDAR 2008, pp. 1–6 (2008)
3. Chui, C.K., Kao, B., Lo, E., Cheng, R.: I/o-efficient algorithms for answering pattern-based aggregate queries in a sequence olap system. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 1619–1628 (2011)
4. Chui, C.K., Kao, B., Lo, E., Cheung, D.: S-olap: an olap system for analyzing sequence data. In: Proceedings of the 2010 International Conference on Management of Data, SIGMOD 2010, pp. 1131–1134 (2010)

5. Chui, C.K., Lo, E., Kao, B., Ho, W.-S.: Supporting ranking pattern-based aggregate queries in sequence data cubes. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, pp. 997–1006 (2009)
6. Gonzalez, H., Han, J., Li, X.: Flowcube: constructing rfid flowcubes for multi-dimensional analysis of commodity flows. In: Proceedings of the 32nd International Conference on Very Large Data Bases, VLDB 2006, pp. 834–845 (2006)
7. Gonzalez, H., Han, J., Li, X.: Mining compressed commodity workflows from massive rfid data sets. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM 2006, pp. 162–171 (2006)
8. Gonzalez, H., Han, J., Li, X., Klabjan, D.: Warehousing and analyzing massive rfid data sets. In: Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, pp. 83–93 (2006)
9. Kaghazian, L., McLeod, D., Sadri, R.: Scalable complex pattern search in sequential data. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 1467–1468 (2008)
10. Li, X., Han, J., Gonzalez, H.: High-dimensional olap: a minimal cubing approach. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, vol. 30, pp. 528–539 (2004)
11. Liu, M., Rundensteiner, E., Greenfield, K., Gupta, C., Wang, S., Ari, I., Mehta, A.: E-cube: multi-dimensional event sequence analysis using hierarchical pattern query sharing. In: Proceedings of the 2011 International Conference on Management of Data, SIGMOD 2011, pp. 889–900 (2011)
12. Liu, M., Rundensteiner, E.A.: Event sequence processing: new models and optimization techniques. In: Proceedings of the Fourth SIGMOD PhD Workshop on Innovative Database Research, IDAR 2010, pp. 7–12 (2010)
13. Livny, M., Ramakrishnan, R., Beyer, K., Chen, G., Donjerkovic, D., Lawande, S., Myllymaki, J., Wenger, K.: Devise: integrated querying and visual exploration of large datasets. *SIGMOD Rec.* 26(2), 301–312 (1997)
14. Lo, E., Kao, B., Ho, W.-S., Lee, S.D., Chui, C.K., Cheung, D.W.: Olap on sequence data. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, pp. 649–660 (2008)
15. Mabroukeh, N.R., Ezeife, C.I.: A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.* 43(1), 3:1–3:41 (2010)
16. Ramakrishnan, R., Donjerkovic, D., Ranganathan, A., Beyer, K.S., Krishnaprasad, M.: Ssql: Sorted relational query language. In: Proceedings of the 10th International Conference on Scientific and Statistical Database Management, SSDBM 1998, pp. 84–95 (1998)
17. Sadri, R., Zaniolo, C., Zarkesh, A., Adibi, J.: Optimization of sequence queries in database systems. In: Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2001, pp. 71–81 (2001)
18. Sadri, R., Zaniolo, C., Zarkesh, A., Adibi, J.: Expressing and optimizing sequence queries in database systems. *ACM Trans. Database Syst.* 29(2), 282–318 (2004)
19. Seshadri, P.: Predator: a resource for database research. *SIGMOD Rec.* 27(1), 16–20 (1998)
20. Seshadri, P., Livny, M., Ramakrishnan, R.: Sequence query processing. In: Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, SIGMOD 1994, pp. 430–441 (1994)
21. Seshadri, P., Livny, M., Ramakrishnan, R.: Seq: A model for sequence databases. In: Proceedings of the Eleventh International Conference on Data Engineering, ICDE 1995, pp. 232–239 (1995)
22. Seshadri, P., Livny, M., Ramakrishnan, R.: The design and implementation of a sequence database system. In: Proceedings of the 22th International Conference on Very Large Data Bases, VLDB 1996, pp. 99–110 (1996)

23. Villafane, R., Hua, K.A., Tran, D.A., Maulik, B.: Mining Interval Time Series. In: Mohania, M., Tjoa, A.M. (eds.) DaWaK 1999. LNCS, vol. 1676, pp. 318–330. Springer, Heidelberg (1999)
24. Wu, E., Diao, Y., Rizvi, S.: High-performance complex event processing over streams. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, SIGMOD 2006, pp. 407–418 (2006)
25. Zhou, B., Jiang, D., Pei, J., Li, H.: Olap on search logs: an infrastructure supporting data-driven applications in search engines. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 1395–1404 (2009)

Data Management for Fingerprint Recognition Algorithm Based on Characteristic Points' Groups

Michał Szczepanik and Ireneusz Józwiak

Abstract. In this paper authors presents data management solutions for biometric system base on firngerprint recognitoin. They compared existing fingerprint recognition data store methods and their own solutions for algorithm wich base on minutes groups. Authors proposed a new algorithm based on distribution minutiaes' groups using selective attention algorithms, which store only base information about minutiaes' positions. The proposed algorithm was compared with existing solutions during analysis of damage fingerprints using false acceptance rate and false rejection rate.

1 Introduction

Fingerprint recognition is one of the most popular biometric techniques. In addition to comparing the quality of fingerprint algorithm, is also very important way to store fingerprint data. The first biometric systems retain the original image, this meant that the normally low-moly encrypted data to be stolen and used to force the system. The main problem of data management in biometric systems is structure of a pattern image. It must be encrypted and also ensure that the decryption will prevent reconstruction of the original fingerprint.

The biggest problem of fingerprint recognition systems is usability. Every day, people are exposed to cuts, wounds and burns, therefore it is important that the algorithms were resistant to this type of damage. Current fingerprint recognitions systems for mobile devices are usually used one of the algorithms like:

- Minutiae Adjacency Graph (MAG),
- Elastic minutiae matching (EMM),

Michał Szczepanik · Ireneusz Józwiak

Wrocław University of Technology, Institute of Informatics Wybrzeże Wyspiańskiego 27,
50-370 Wrocław, Poland

e-mail: [michal.szczepanik, ireneuszl.jozwiak}@pwr.wroc.pl](mailto:{michal.szczepanik, ireneuszl.jozwiak}@pwr.wroc.pl)

- Delaunay Triangulation (DT),
- Pattern-Based Templates (PBT).

Most popular algorithms based on local and global structures are represented by graphs like in MAG. In this types of algorithm first local structures are used to find corresponding points to align feature vector, then global structures are matched [4]. This type of algorithm was used by Ross et al. [16], He and Ou [6]. They also use thin-plate spline (TPS) model to build an average deformation model from multiple impressions of the same finger. Owing to iteratively aligning minutiae between input and template impressions, a risk of forcing an alignment between impressions originating from two different fingers arises, and leads to a higher false accept rate. Typically, minutiae matching has two steps:

- registration aligns fingerprints, which could be matched, as well as possible
- evaluation calculates matching scores using a tolerance box between every possibly matched points (minutiae) pairs

In MAG algorithm each minutiae is described by 3 or 4 parameters $v = (x, y, T, \Theta)$, where x and y are the coordinate, T is an optional parameter specifying the type and Θ which determines orientation of minutiae. In addition, defined edges connecting the two points representing the minutiae of each edge is defined as follows $e = (u, v, rad, r_c, \theta)$, where u, v are nodes (minutiae) initial and marginal, rad it is Euclidean distance between minutiae, r_c determines the distance by the number of ridges between minutiae, and θ is the angle between the edge and the axis x .

The EMM algorithm typically uses only global matching, where each point (minutia) which has a type, like end point or bifurcation needs to match to with a related point in second finger print image. Base on elastic deformations which is are used to tolerate minutiae pairs that are further apart because of plastic disrotations, and therefore to decrease the False Rejection Rate, so in most popular algorithms authors increase the size of bounding boxes [13] to reduce this problem, but they get higher False Acceptation Rate (FAR) as a side effect. In this type of algorithm for elastic match also TSP [1] can be used, which provides better performance than only one parameter of deformation. In EM algorithm data, about characteristic points, are storage in analogical way like in MAG algorithm. Each minutiae is represented by $p = (x, y, T, \Theta)$, where x and y are the coordinate, T is an optional parameter specifying the type and Θ which determines orientation of minutiae.

The Delaunay Triangulation algorithm [2, 14] based on triangulation connects neighboring minutiae to create triangles, such that no point (minutia) in P is inside the circumcircle of any triangle in $DT(P)$. In this algorithm, the characteristic point information is stored in the same way as in the EMM, the only difference is the method of processing which based on triangulation. Unfortunately, just as minutiae adjacency graph algorithm is not resistant to injury of physical fingerprint.

Pattern based algorithms [3] compare the basic fingerprint patterns (like arch, whorl, and loop between a previously stored template and a candidate fingerprint. Those algorithms requires that the images be aligned in the same orientation and in the same scale. To do this, the algorithm finds a central point in the fingerprint

image and centers on that, and after that, scales to the same size of fingerprints ridge. In a pattern-based algorithm, the template contains the type, size, and orientation of patterns within the aligned fingerprint image. The candidate fingerprint image is graphically compared with the template to determine the degree to which they match. Due to the storage of the original picture for algorithm there is a high risk that this image can be read from the memory card reader or fingerprints database.

2 Fingerprint Recognition Algorithm Based on Minutes Groups

The proposed solutions, in contrast to other algorithms, are more resistant to damage.

2.1 Fingerprint Recognition Algorithm Based on Minutes Groups

For older low-resolution readers it is required to detect the areas of correct scanning of the fingerprint. First step of image analysis is the search for the imprint area including the exclusion of areas containing significant damage. Fingerprint image is represent by a gray scale image that defines the area of forced application fingerprint for the reader.

$$I_{fp}(i, j) = \langle 1, 255 \rangle \quad (1)$$

The operation that converts a grayscale image into a binary image is known as binarization. We carried out the binarization process using adaptive thresholding. Each pixel is assigned a new value (1 or 0) according intensity mean in a local area and the parameter t_g which excludes poorly read fingerprint areas from the analysis.

$$B_{fp}(i, j) = \begin{cases} 1 & \text{for } I_{fp}(i, j) \geq t_g \\ 0 & \text{for } I_{fp}(i, j) < t_g \end{cases} \quad (2)$$

The last step is creating the fingerprint mask based on the binarized image. The Mask for the area of a square (X, Y) , which size is 2.5 wide edges, is determined by two parameters p_{lo} , which is a limitation that excludes areas with an insufficient number of pixels describing the image, and p_{hi} excludes blurred areas, such as moist.

$$F_{fp}(X, Y) = \begin{cases} I_{fp}(X, Y) & \text{for } \sum_{i \in X} \sum_{j \in Y} B_{fp}(i, j) \geq p_{lo} \wedge \sum_{i \in X} \sum_{j \in Y} B_{fp}(i, j) \leq p_{hi} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Created mask is used for finding the most damages area in the fingerprint image.

2.2 *Detecting Features and Levelling of the Damage in Segmentations*

Standard leveling of damage is carried out by calculating the variance of points and the analysis of brightness. Based on these two parameters, the frequency of furrows is calculated. Which is used for each fingerprint image.

After applying Gabor filter [12] to highlight the pits and valleys, it uses segmentation, in accordance with its size 2.5 width of segment furrow, the image is redrawn.

After that process fingerprints are continuous and lint. In contrast to the literature, the algorithm does not require additional transformations to find the minutiae, such as converting all the width of 1px furrows. It does not require information about the orientation of minutiae, it only requires the data about its position. Therefore, the resulting image is used to find the edge - the minutiae are located at the intersection of the edge of the furrows.

The problem of fingerprint recognition is a complex process, even in laboratory conditions, therefore, if used as the system to control access to the mobile devices, it should be insensitive to certain natural changes or damages in physical structure of fingerprints, which can include: incomplete fingerprint, fingerprint parts which can be injured or burned, rotation, blurred or partly unreadable.

In order to detect the areas most sensitive to damage, we use neural network with selective attention technique. This type of neural network is more like an analysis done by a human. This allows us to create a mask of areas vulnerable to damage.

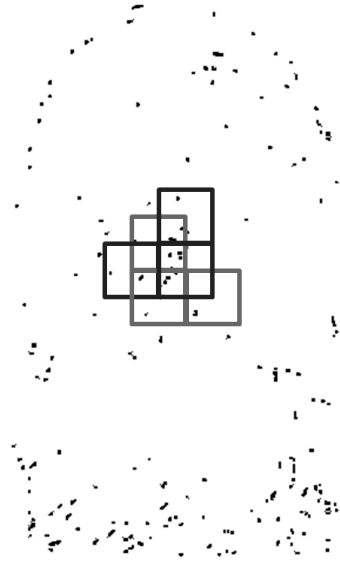
We created 15 different masks, broken down by the type of fingerprints core also known as fingerprint patterns (arch, whorl and loop) and the type of finger (thumb, index finger, middle finger, ring finger, small finger). Basing on this mask we created a filter which we use to compare fingerprints where specific minutiae are weighted in the decision process and their score is based on the location on the fingerprints.

3 Fingerprints Comparison

Minutiae image is divided into segments, each segment corresponding minutiae group is described by parameters (x, y, nom) , where x and y are the coordinates, and nom determines the number of minutiae in the group. Additionally, one implementation uses an additional parameter specifying the probabilities of damage in a given segment which is estimated by a neural network. , Based on the distribution of areas rejected by the mask described by the formula. The last step is to create a matrix of Euclidean distances between the groups.

When comparing the use of two parameters: dx - the distance, the difference between groups in the pattern and fingerprint test px - the threshold probability of damage (determined by whether the group is under consideration in the analysis).

Fig. 1 Fingerprint divided into segments. (Source: own work).



When comparing, the groups are divided according to the weight that defines the number of minutiae in the group and selective attention (SA) algorithms, which are based on probabilities of damage in a group segment. This provides quick verification of whether the analyzed fingerprint is consistent with the pattern.

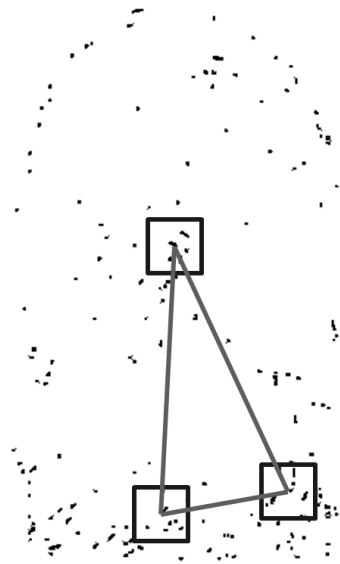
4 The Quality of the Algorithms

For test authors used real fingerprints. Due to the nature of work performed by a group of testers, they where exposed to frequent damage fingerprints. All four algorithms were compared using the fingerprint database of 120 different fingerprints which had 8 samples of each fingerprint. The database contained 10% of the fingerprints with damage which were mostly cuts and burns, so it simulated the most

Table 1 The result of experiment using real fingerprits

	FAR	FRR
MAG	1.58%	0.95%
EMM	2.35%	2.65%
PBTA	0.35%	8.52%
MGM64	8.45%	0.10%
MGM32	3.10%	0.10%
MGM32_SA	0.30%	0.10%

Fig. 2 Detecting relations between minutiae groups. (Source: own work)



frequently encountered damage types in daily life. The first test compares the existing algorithms with the proposed one.

Most algorithms are almost immune to physical damage of fingerprints. Also, the proposed one has proven to have a very dangerous level of False Acceptation Rate. After applying the selective attention algorithm, fingerprint recognition algorithm improved their performance and reliability. The Proposed algorithm has been developed in such a way, that it uses the property of a damage map, so its results have improved the most.

Second test was done using FVC2004 [12] fingerprints databases. For each of four database a total of 120 fingers and 12 impressions per finger (1440 impressions) were gathered. Unfortunately, most of the publicly available database of fingerprints does not include the problem of physical damage, so additionally on each sample have been generated small damage such as cuts and burns.

Table 2 The result of experiment using FVC2004 database

	FAR	FRR
MAG	0.82%	0.65%
EMM	1.23%	1.15%
PBTA	0.15%	1.73%
MGM64	6.90%	0.65%
MGM32	3.66%	0.42%
MGM32_SA	0.35%	0.12%

5 Data Management

Developed algorithm is based on minutiae groups where each group is basically represented by the coordinates - x, y and the number of minutiae - nom contained in the group. Group covers an area equal to 2.5 the width of the furrow, its coordinates are in the middle of the square which bounding this area. Number of minutiae in the group determines its priority, additionally stored parameter describing the probability of damage - p_d in the area represented by the group. In conclusion the group is defined as follows:

$$M_{group} : \{x, y, nom, p_d\} \quad (4)$$

Based on these data creates a matrix of Euclidean distances between the groups. Data on the characteristic point is limited to its weight (nom) and the probability of damage p_d . Finally we obtain:

$$M_{group}(I) : \{nom_I, (p_d)_I\} M_{group}(I, J) : dist(M_{group}(I), M_{group}(J)) \quad (5)$$

Where $dist(M_{group}(I), M_{group}(J))$ is Euclidean distances between the group I and J. Data stored for analysis to prevent reproduction of the original fingerprint image. Additional storage parameters to estimate the damage Allows you to better match fingerprints in the event of damage.

6 Conclusion

The proposed algorithm enables the identification of fingerprints in places where they are exposed to frequent damage. Way to store fingerprint data makes it impossible to recreate the original structure. It provides information about the minutiae clusters and their frequency. In the future work, data management will be extended by storing information about events of damages which were recognition ed by identifies fingerprints algorithm.

Acknowledgements. This experiment was inspired by interactions with the Bitbar company and was co-financed by the European Union under the European Social Fund.

References

1. Bazen, A.M., Gerez, S.H.: Fingerprint matching by thinplate spline modelling of elastic defromations. Pattern Recognition (2003)
2. Bebis, G., Deaconu, T., Georgiopoulos, M.: Fingerprint Identification Using Delaunay Triangulation. In: IEEE ICIS, pp. 452-459 (1999)

3. Cappelli, R., Lumini, A., Maio, D., Maltoni, D.: Fingerprint Classification by Directional Image Partitioning. *IEEE Transactions on Pattern Analysis Machine Intelligence* 21(5), 402–421 (1999)
4. Chikkerur, S., Cartwright, A.N., Govindaraju, V.: K-plet and Coupled BFS: A Graph Based Fingerprint Representation and Matching Algorithm. In: Zhang, D., Jain, A.K. (eds.) *ICB 2005. LNCS*, vol. 3832, pp. 309–315. Springer, Heidelberg (2005)
5. Grzeszyk, C.: Forensic fingerprint examination marks. Wydawnictwo Centrum Szkolenia Policji, Legionowo (1992) (in Polish)
6. He, Y., Ou, Z.: Fingerprint matching algorithm based on local minutiae adjacency graph. *Journal of Harbin Institute of Technology* 10(05), 95–103 (2005)
7. Huk, M., Szczepanik, M.: Multiple classifier error probability for multi-class problems. *Maintenance and Reliability* 3, 12–17 (2011)
8. Hicklin, A., Watson, C., Ulery, B.: How many people have fingerprints that are hard to match, NIST Interagency Report 7271 (2005)
9. Hong, L., Wan, Y., Jain, A.K.: Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 777–789 (1998)
10. Indovina, M., Uludag, U., Snelick, R., Mink, A., Jain, A.: Multimodal Biometric Authentication Methods: A COTS Approach. In: *Proc. MMUA* (2003)
11. Jain, A.K., Ross, A., Nandakumar, K.: *Introducing to biometrics*. Spinger (2011)
12. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: *Handbook of Fingerprint Recognition*, 2nd edn. Springer (2009)
13. Pankanti, S., Prabhakar, S., Jain, A.K.: On the individuality of fingerprints. In: *Proceedings of Computer Vision and Pattern Recognition, CVPR* (2001)
14. Parziale, G., Niel, A.: A Fingerprint Matching Using Minutiae Triangulation. In: Zhang, D., Jain, A.K. (eds.) *ICBA 2004. LNCS*, vol. 3072, pp. 241–248. Springer, Heidelberg (2004)
15. Ratha, N.K., Govindaraju, V.: *Advances in Biometrics: Sensors, Algorithms and Systems*. Springer (2007)
16. Ross, A., Dass, S.C., Jain, A.K.: A deformable model for fingerprint matching. *Pattern Recognition* 38(1), 95–103 (2005)
17. Ross, A., Nandakumar, K., Jain, A.K.: *Handbook of Multibiometrics*. International Series on Biometrics. Springer (2011)
18. Shimooka, T., Shimizu, K.: Artificial Immune System for Personal Identification with Finger Vein Pattern. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) *KES 2004. LNCS (LNAI)*, vol. 3214, pp. 511–518. Springer, Heidelberg (2004)
19. Szczepanik, M., Szewczyk, R.: Fingerprint identification algorithm. *KNS* 1, 131–136 (2008) (in Polish)
20. Wayman, J.L., Jain, A.K., Maltoni, D., Maio, D.: *Biometric Systems. Technology, Design and Performance Evaluation*, 1st edn. Springer (2005)

Data Prefetching Based on Long-Term Periodic Access Patterns

Dmitri Vasilik

Abstract. Data prefetching is a technique allowing to retrieve data which will be most likely needed in a near future, before the actual demand. Considerable research was devoted to this technique, however, it is typically based on short-term data access patterns. We propose to predict future accesses based on long-term periodic pattern mining. Human activity in many areas, and thus many real-world business processes appear to have natural periods: they may have day, week and/or month periods. Discovering of such periods in I/O (or higher level) activity logs should allow to build a prefetch predictor, which is aware of data accesses not only in a near future, but in a far future perspective as well, and thus able to make more reasonable prefetch decisions. In this work we investigate the algorithm for mining periodic long-term access patterns, and discuss issues involved in building a prefetch system, which integrates predictor based on discovering these patterns with a prefetch cost model.

1 Introduction

The gap between CPU performance and I/O performance was great 40 years ago and is not becoming any smaller. Among techniques designed to overcome this problem are multithreaded execution, which hides memory and disk latencies, and caching which allows applications to avoid expensive accesses to lower level data storage. Another technique is data prefetching which allows to retrieve the data before it is actually requested.

For instance, disk controller reads a number of sequential data blocks along with the requested one. It allows to serve a number of consequent application read request in a single disk read. This simple idea is effectively used in software as well:

Dmitri Vasilik

Saint Petersburg State University, 7-9, Universitetskaya nab., St.Petersburg, Russia

e-mail: Dmitri.Vasilik@gmail.com

file systems and DBMSs do essentially the same. Advanced data storages employ adaptive prefetch: storage operating environment decides how much data to prefetch based on analysis of a client data access pattern. Storage uses aggressive prefetch, as soon as it detected sequential access pattern, otherwise less aggressive.

Under this approach the prefetch is based on heuristic guess that consequent data accesses will be sequential. Although this approach may be very effective in some cases, it does not capture more complex access patterns and its benefits are limited by the notion of "sequential" in the corresponding abstraction, e.g. file system cannot continue prefetching after the end of file, because it does not have a concept of a "next file".

Another approach to data prefetching is to learn file relationship. File accesses are not random, moreover file accesses usually follow previous patterns with high probability. For instance, if strong relationship between files A, B and C exists, these three files can be fetched together when one of them is accessed. A simple last successor model, which predicts that file B will be accessed after A, if it was accessed after A last time A was accessed, may be very effective in some workloads. [5] reported that on real file system traces last successor correctly predicted 72% of file accesses.

[2] presents Recent Popularity model based on last successor. The idea is to select the best of k previous successors. Best is chosen as seen in successor list maximum number of times. Predictor output has a confidentiality attribute, so the system, which uses this predictor, may ignore predictions with low probability.

[4] presents a simple model for learning file relationship based on prediction graph. Files represent vertexes in a graph, if two files were accessed consequently (within a lookahead period) edge between them is drawn. Every edge is marked with a number of occurrences of access sequence in a window. The probability, that file B will be accessed after A, is calculated as a relationship between the weight of A-B edge divided by sum of weights of edges leaving A.

A number of more complex models based on graphs, heuristics and finite multi-order context modeling was proposed for studying file relationship. Of special interest is [6], which presents a C-Miner, an algorithm which uses frequent sequence mining to discover disk-block correlations. Authors reported block correlations mined by C-Miner from the first 3 days of Cello-92 workload to be still effective for the next 4 days. Authors noted that as block correlations found are relatively stable, there is no need to run C-Miner continuously to update block correlations.

Based on idea of pattern stability, we propose a future-aware algorithm based on long-term periodic pattern mining. Human activity in many areas appears to be periodic, thus I/O activity in many real-life workloads is periodic as well. Consider a business analyst that starts preparing a weekly report at Friday morning. He starts query scanning tens of gigabytes of data and needs to wait for significant amount of time, before data warehouse finishes query processing. If we know that he do this on a regular basis, i.e. he starts every Friday between 9 and 10 AM, we could prepare the data he will most likely need before he requests it.

Another example is a freight ship company which owns and operates the freight ships, which transport trucks. Ships departure conforms to schedule, which does not change for months. This schedule has day, week and/or month periods. Before ship may sail, manifest listing all cargo has to be printed. If we prefetch the data needed for manifest, the company will save expensive time in port.

However, in both examples "important" activity, which people may wait to conclude, may be interleaved with some background activity. Consider the second example with the cargo company. Execution of the query gathering data for manifest may be interleaved with tens of thousand reservation and registration operations.

2 Pattern Mining

Recently our research group has finished two projects directly or indirectly addressing problems arising in building prefetch system based on long-term periodic pattern mining.

The main goal of the first project was to develop an algorithm, which allows to find periodic business processes in DBMS activity logs. Classical frequent pattern mining algorithms do not work well on this data, because of a large amount of "noise". Actually they are not applicable to our task, since the main measure of pattern importance in these algorithms is a frequency of pattern occurrence. A comprehensive review of advances in these field since Agrawal's work ([1]) along with the discussion of pattern importance measures may be found in [8].

Algorithm presented in this project is executed in three phases: first the data is cleaned, at the second phase algorithm finds groups of correlated queries (business processes), after that data corresponding to occurrences of these groups in logs is mined for periods.

At the first phase two types of activity are filtered from logs. First, the requests which appear too often and too rare are removed from the logs. Requests appearing too often correspond to reservation activity in example about freight ship company. This activity is of no interest, because it consists of single requests, which do not constitute a business process. Second, "constant" activity is filtered based on minimum execution time variance threshold. Permanent activity may contain some business processes, however, conventional caching and prefetching algorithms should cope with it well.

The second, mining phase, is based on idea that requests which belong to a single business process should be close to each other in time. Analysed log consists of "snapshots", each containing a hour of activity. The interconnectedness measure for two requests q_1 and q_2 presented in this project was given in terms of the number $s(q_1, q_2)$ of "snapshots", in which both requests appear and the number of "snapshots" $s(q_1)$ and $s(q_2)$, where q_1 and q_2 appears accordingly.

$$I = \frac{s(q_1, q_2)}{s(q_1) * s(q_2)}$$

Formula for request group is similar. The group of requests is considered to be a business process if its interconnectedness measure exceeds predefined threshold.

Once request groups intercorresponding to business processes are found, the third phase is started. During this phase algorithm builds a binary vector for each group of correlated requests. This vector has 1 on i -th position, if i -th snapshot contains requests of the group, otherwise 0. Periods are then found in this vector using cyclic association rule detection techniques.

Algorithm may be naive in some sense, yet it has shown 65% precision of business process detection on logs of an industrial database. Pattern confidence for this algorithm may be derived from requests interconnectedness measure and relaxations made at the third phase to find periods.

One of the most important decisions made in this work was to separate the search for groups of related requests from the mining of periods. So the problem of finding periodic business processes is effectively divided into two parts. While the second part is quite simple, the first two phases of the algorithm need to be revisited and generalized. At this point the algorithm for finding related request groups dramatically lacks generality. All threshold values, used in filtering and main mining phases were selected manually and are highly depended on the features of the data. In fact, significant amount of time was spent investigating the structure of activity of testing data set and experimenting with different parameter sets.

3 Prefetch Cost

I have not participated in the development of the presented algorithm, yet, I do participate in its further development now. While my colleagues were working on this algorithm, I was involved in another project, which was aimed to develop a detailed simulation performance model for three distributed data warehousing systems working under TPC-H workload. This work was focused on performance of sequential reads, which constitute the significant part of workload generated by TPC-H queries. The insights on enterprise data storage performance issues gained in this project should be very useful for deciding when to prefetch data.

The first problem to solve is that, if the system load is too heavy, prefetching can add significant overhead to demand requests. A solution would be to adjust pattern confidence threshold according to current system load. For instance, if system utilization is higher than 80% we do not prefetch data with pattern confidence less than 90%. As system load drops down we decrease the confidence threshold.

Though, for complex systems such as enterprise data storages it may be quite hard to estimate the negative impact of prefetching a particular data element on demand requests executed at the moment. Contemporary enterprise data storages have quite complex hardware architecture, and are equipped with tens of multi-core CPUs, hundreds of gigabytes of RAM and serve I/O requests to hundreds of disks. At the disposal of such systems are not only cache, CPU power and disk arm heads, but also networks connecting disk drive enclosures to data storage engines.

For instance, if data storage is used under OLTP workload the performance may be limited by disks, but under OLAP workload the back-end networks connecting data storage engines to disks may become highly saturated, while disks are underutilized.

Additional overhead introduced by a prefetch request would depend on hardware resources involved and whether the utilization of these resources is critical to the system performance. To estimate this overhead we may try to identify hardware resources that appear on the data path for the prefetch request and evaluate impact on the service time of demand requests, currently being executed at each resource using utilization statistics.

Usage of cache, which has a limited size, for prefetched data blocks is another important issue which may have significant impact on the system performance. [3] has shown that aggressive prefetch policy (with absolutely correct prefetch guesses and optimal block replacement choices) may be outperformed by no-prefetch policy in some cases. Prefetching file blocks into cache may be harmful, even if these blocks will be accessed in a near future, because the cache in turn evicts blocks which might be referenced in a near future as well. The article presents four rules, that an optimal prefetching and caching strategy must follow.

[7] advocates integrating cache and prefetch strategies as well. In this work authors investigate the tradeoff between satisfying competing needs of a system with limited memory size: for prefetching hinted blocks and caching demanded blocks. Authors evaluate the cost of giving an additional buffer to prefetch cache and taking it from on-demand data cache under a set of strong simplifying assumptions about the system. Yet, the framework for cache management developed and implemented in this work was shown to be quite effective in a number of various workloads.

To sum up, the cost function we have to develop should depend on three factors: additional overhead incurred by reading of data element, caching cost and predictor confidence.

Another question to be considered is at what moment should prefetch be started? If the prefetch is started too early, there is a chance the prefetched data will be displaced, so prefetch requests must be executed in a timely manner. The time to read a data should depend on the performance and utilization of resources on the data path, however, it may be estimated with statistical data.

4 System Implementation

The basic choice in the implementation of a prefetch system based on long-term data access patterns is at what level should data mining module reside. The predictor module may reside in an operational system or a data storage ("block box" approach), or in an application ("white box" approach). For instance, "black box" approach is used by C-Miner ([6]): C-Miner infers file system block correlations without any assumption or modifications to storage front-end. Block correlations are discovered fully transparently for applications by only observing access sequences. Cache management framework, presented in [7], represents an example of "white

box” approach: application gives disclosure hints to file system via I/O Control interface (`ioctl` function). The basic idea of this approach is that application knows it future demands better.

While the ”black box” approach appears to be much more universal, it limits the benefits of data prefetching. Under this approach data mining algorithm may operate only with low-level I/O activity, i.e. file or block accesses. On this abstraction level it is impossible to find some important patterns, which may constitute a large share of overall system workload. Consider a freight ship company example once again. The data generated by registration queries goes to the same table, but not to the same place in a file system. Thus, the query building manifest actually scans the different files or disk blocks each time. The data in the file system, which was already scanned, will not be scanned in any consequent requests.

The solution to this problem is to place a prefetch predictor inside a DBMS. Under this approach, the data mining algorithm should discover such a business process, because the queries employed are essentially the same. Moreover, DBMS may provide a prefetch module with information about files, storing the data of tables which will be accessed by a request. However, this information may be still insufficient for efficient prefetch, since the data prefetch module knows nothing about what part of the table should be retrieved, and tables may be very large. To know what file blocks would be read by a query, a query execution plan is needed. However, an actual query plan can not be built without query parameters, which are not known in advance. Guessing query parameters based on previous values may be very tricky.

5 Conclusion

In this work we propose a data prefetch algorithm based on detection of business processes in activity logs. To sum up all previously said, we have three main directions for future work:

- **Develop more general mining algorithm.** Algorithm developed in previous work is quite naive and demands a set of data depending parameters. We believe it is possible to build an algorithm general enough, and yet showing high precision and recall values on our data set.
- **Build a simple prefetching system.** Design and implement a simple system for prefetching patterns with fixed confidence threshold based on mining patterns from file system or DBMS traces.
- **Develop a cost model for prefetching based on system load.** Design a system which will be able to decide would it be beneficial to prefetch data for a particular pattern or not.

The task is quite tough, yet challenging: the reader may have noted that currently there are many opened questions and unsolved problems, concerning both system implementation and cost model. Mining algorithm needs further research as well. However, we hope the PhD work would be finished in time with some working system prototype.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco (1994)
2. Amer, A., Long, D.D.E., Paris, J.-F., Burns, R.C.: File access prediction with adjustable accuracy. In: 21st IEEE International Proceedings of the Performance, Computing, and Communications Conference, PCC 2002, pp. 131–140. IEEE Computer Society, Washington, DC (2002)
3. Cao, P., Felten, E.W., Karlin, A.R., Li, K.: A study of integrated prefetching and caching strategies. In: Proceedings of the 1995 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 1995/PERFORMANCE 1995, pp. 188–197. ACM, New York (1995)
4. Griffioen, J., Appleton, R.: Reducing file system latency using a predictive approach. In: Proceedings of the USENIX Summer 1994 Technical Conference on USENIX Summer 1994 Technical Conference, USTC 1994, vol. 1, p. 13. USENIX Association, Berkeley (1994)
5. Kroeger, T.M., Long, D.D.E.: The case for efficient file access pattern modeling. In: Proceedings of the The Seventh Workshop on Hot Topics in Operating Systems, HOTOS 1999, p. 14. IEEE Computer Society, Washington, DC (1999)
6. Li, Z., Chen, Z., Srinivasan, S.M., Zhou, Y.: C-miner: Mining block correlations in storage systems. In: Proceedings of the 3rd USENIX Conference on File and Storage Technologies, FAST 2004, pp. 173–186. USENIX Association, Berkeley (2004)
7. Patterson, R.H., Gibson, G.A., Ginting, E., Stodolsky, D., Zelenka, J.: Informed prefetching and caching. *SIGOPS Oper. Syst. Rev.* 29(5), 79–95 (1995)
8. Yang, J., Wang, W., Yu, P.S.: Infominer: mining surprising periodic patterns. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2001, pp. 395–400. ACM, New York (2001)

E-ETL: Framework For Managing Evolving ETL Processes

Artur Wojciechowski

Abstract. Data warehouses integrate external data sources (EDSs), which very often change their data structures (schemas). In many cases, such changes cause an erroneous execution of an already deployed ETL workflow. Structural changes of EDSs are frequent, therefore an automatic reparation of an ETL workflow, after such changes, is of a high importance. This paper presents a framework for handling the evolution of an ETL layer – *E-ETL*. Detection of changes in EDSs causes a reparation of the fragment of ETL workflow which interacts with the changed EDS. The proposed framework was developed as a module external to an ETL engine, accessing the engine by means of API. The innovation of this framework are algorithms for semi-automatic reparation of an ETL workflow.

1 Introduction

A data warehouse (DW) is usually created to integrate multiple heterogeneous, distributed, and autonomous external data sources (EDSs). Such integrated data can be used for analysis, called On-Line Analytical Processing (OLAP). One of the DW elements is an ETL process which extracts data from EDSs, transforms data into a common data model, cleans data (removes missing, inconsistent, and redundant values), integrates data, and loads them into a DW. An inherent feature of EDSs is their evolution in time with respect not only to their contents (data) but also to their structures (schemas). Since changes of EDSs structure may cause erroneous execution, after every such a change, an ETL process must be redesigned and redeployed. Frequent manual modifications of an ETL process are complex, prone-to-fail, and time-consuming. Hence, it is of a high importance to develop methods for handling

Artur Wojciechowski

Poznań University of Technology, Institute of Computing Science, Poznań, Poland

e-mail: artur.wojciechowski@cs.put.poznan.pl

<http://calypso.cs.put.poznan.pl/projects/e-etl/>

structural changes of EDSs and managing the evolution of the ETL process. So far research community did not give much attention to the evolution of the ETL layer [2, 5].

Paper Contribution. This paper contributes a framework, called *E-ETL*, for: (1) detecting structural changes of EDSs and (2) handling the changes at the ETL layer. Changes are detected either by means of Event-Condition-Action (triggers) mechanism or by means of comparing two consecutive EDS metadata snapshots. Detection of the EDS schema change causes a reparation of the ETL activities that interact with the changed EDS. The reparation of the ETL activities is guided by several customizable reparation algorithms. The proposed framework was developed as a module external to an ETL engine. Communication between *E-ETL* and the ETL engine is realized by means of the ETL engine API. The framework is customizable and it allows to: (1) work with different ETL engines that provide API communication, (2) define the set of detected structural changes, (3) modify and extend the set of algorithms for managing the changes, (4) define rules for the evolution of ETL processes (5) present to the user the impact analyses of the ETL workflow, (6) store versions of the ETL process and history of EDS changes. The framework has a graphical user interface for visualizing ETL processes.

The main conception of the presented solution is based on some ideas presented in [2]. The possibility of using *E-ETL* is extended by co-operation with external ETL tools. Moreover *E-ETL* defines a system of detecting structural changes in EDSs and extends algorithms for managing the changes.

Paper Organization. The paper is organized as follows. Section 2 presents the concept of the *E-ETL* framework. Section 3 describes the *E-ETL* internal metamodel. Section 4 introduces reparation algorithms. Section 5 overviews detected schema changes and evolution rules. Section 6 outlines research related to the topic of this paper. Section 7 summarizes the paper and outlines issues for future development.

2 Concept of the E-ETL Framework

The *E-ETL* project focuses on developing a method and a framework to support the semi-automatic evolution of ETL process. In particular, the research and development focus on: (1) the development of a prototype architecture, called *E-ETL* that will be able to co-operate with a leading commercial and open source ETL development environments, (2) a graphical interface for visualizing ETL processes, (3) tools for detecting structural changes and propagating them into an ETL layer, (4) a language for defining rules for the evolution of ETL processes, (5) a method for checking the validity of an evolved ETL process, (6) a metamodel for storing versions of ETL processes.

E-ETL is designed to co-operate with ETL development environments (currently the Microsoft SQL Server Integration Services is supported). To this end, *E-ETL*

is an external system to an ETL development environment. *E-ETL* connects to a development environment by means of API.

E-ETL analyses the design of an ETL process which is defined in an ETL development environment, and on the basis of this project an internal model of the ETL process is created. Next, an ETL designer defines a set of rules that specify how the ETL process should evolve in response to the detected changes. Then, when *E-ETL* detects structural changes in an EDS, it proposes semi-automatically (in some cases automatically) the modifications of the ETL process. After a user's acceptance of the changes, *E-ETL* applies them to the ETL process in the ETL development environment.

3 Internal Metamodel

Different ETL development environments may use different data models. Therefore, the *E-ETL* framework uses its own internal data model that permits to unify work with external ETL systems. In this model, an ETL process is represented as a directed graph. Each activity in the ETL process is presented as *SuperNode*. *SuperNode* consists of *Nodes* that represents input and output parameters of an ETL activity. An input parameter can be a table attribute that the activity reads, a node in XML structure, or a column in a spreadsheet. Dependencies between nodes are determined by edges between nodes. So, if there is a directed edge from node *A* to node *B*, then this means that node *B* depends on node *A*. Such model permits to do impact analyses. The impact analyses mark the parts of an ETL process that has to evolve as the result of structural changes in EDSs. These analyses are done by selecting all nodes succeeding the nodes that have been changed (nodes that describe EDS attributes that have been changed). To make the internal metamodel more readable and organized *SuperNodes* can be grouped into *GroupNodes*. *GroupNodes* also can be grouped into *GroupNodes*. This mechanism of grouping allows user to work on different levels of details.

Figure 1 presents an example of an internal metamodel. It shows a fragment of an ETL process that reads the *People* table and splits read data basing on the *Age* attribute. In the next step *People* tuples are joined with data read from *Addresses* table and *addresses.csv* file. *SuperNodes* that represents ETL activities are depicted in the figure as labeled boxes (e.g., *SQL query (1)*, *Conditional split*). Attributes inside boxes (e.g., *Id*, *Street*, *Name*, *Status*) are *Nodes* and they represent ETL activity parameters (input or output).

Exemplary *SuperNode* – *SQL query (1)* defines an activity that is described as an SQL query that selects tuples with *Country* equal to 'Poland'. *Id*, *Street* and *City* output parameters depend respectively on *Id*, *Street*, and *City* input parameters. The SQL query that defines the exemplary activity contains also *WHERE* clause (*Country='Poland'*). Therefore modification or removal of *Country* input parameter would influence the result of the query. Therefore all output parameters also depend

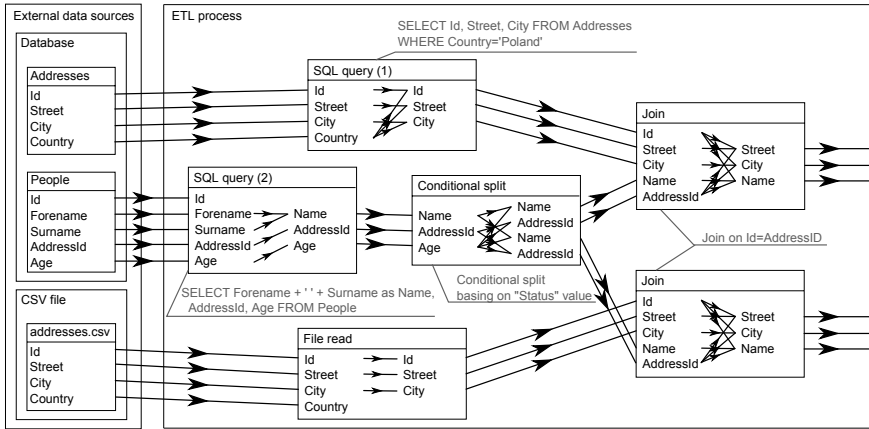


Fig. 1 Internal metamodel example

on *Country* input parameter. The dependencies are shown as directed edges between attributes.

4 Reparation Algorithms

The detection of changes in an EDS fires the execution of algorithms that adapt an ETL process to the detected changes. These algorithms have been categorized as follows: *Defined rules*, *Replacer*, and *Alternative scenarios*. An ETL designer can specify algorithms that are supposed to be used to modify the ETL process, their priorities, and their parameters.

The *Defined rules* algorithm applies evolution rules, defined by a user, to particular elements of an ETL process. For each element (*Node*, *SuperNode*, or *GroupNode*) of an ETL process, a user can define whether this element is supposed to propagate the changes, to block them, to ask a user, or to fire action specific to a detected change.

The *Replacer* algorithm is based on the solution presented in [5]. For each element of an ETL process, a user can define whether this element can be replaced by other element. This replacement can be done when the element has been removed due to changes in EDS.

The *Alternative scenarios* algorithm repairs an ETL process basing on the fact that similar EDSs are usually processed in the same way and also the same changes on similar EDSs should be handled in the same way. Therefore, when the structure of one of the EDSs has changed, then the *Alternative scenarios* algorithm tries to find another EDS with a similar structure. After finding a similar EDS, operations related to both EDSs (the changed EDS and the similar one) are analyzed. This

analysis provides information about differences between an ETL process fragment affected by the detected change and an ETL process fragment related to the similar EDS. Basing on this information, the *Alternative scenarios* algorithm proposes modifications of the ETL process fragment affected by the detected change. Since the history of the ETL process evolution and history of the EDSs changes are stored in the system, the *Alternative scenarios* algorithm can search not only in the current version on an ETL process, but also in their previously used versions. Such functionality may be useful when some changes will be undone in EDSs or changes will be done gradually in sequential EDSs.

5 Monitored Structural Changes

As mentioned before, *E-ETL* detects changes in EDSs either by comparing two successive snapshots of an EDS's metadata, or by the mechanism of triggers (if such triggers are supported and allowed to be installed in an EDS). Changes that can be detected are divided into two groups: *Collections* changes and *Collection element* changes. A *Collection* defines a set of tuples. *Collection elements* define elements of the tuple. Database table, spreadsheet, branch in XML are examples of *Collections* and respectively database table column, column in spreadsheet, node in XML are examples of *Collection elements*. Five changes can be distinguished for *Collections*: (1) *Add*, e.g. a new table addition in a database, (2) *Delete*, e.g. a deletion of a spreadsheet in Excel file, (3) *Rename*, e.g. a change of a file name, (4) *Split*, e.g. a partition of a table, (5) *Merge*, e.g. a merger of partitioned tables. Also five changes can be distinguished for *Collection elements*: (1) *Add*, e.g. a new column addition to a database table, (2) *Delete*, e.g. a deletion of a column in a spreadsheet, (3) *Rename*, e.g. a change of a node name in XML, (4) *Type*, e.g. a change of a column type form numeric to string, (5) *Length*, e.g. a change of a column type length from char(4) to char(8).

All of the mentioned changes are handled by our framework at the level of *SuperNode* (an ETL activity). We adopt a similar solution to the one presented in [2]. On each *GroupNode*, *SuperNode*, or *Node* for every type of change a user can define one of five evolution rules: *Inherit*, *Propagate*, *Block*, *Ask* or *Action*. User can also define default behavior for an element by setting appropriate rule for *Any change*.

The *Inherit* rule means that the rule should be inherited from *Any change*. If the *Inherit* rule is set on *Any change* this means that the rule should be inherited from an enclosing element (for *Node* it is *SuperNode*, for *SuperNode* it is *GroupNode*, and for *GroupNode* it is enclosing *GroupNode*). The *Propagate* rule instructs that the detected change should be propagated through the ETL activity (*SuperNode*). Both, input and output attributes of the activity should be modified accordingly to the change and information about the change should be passed to next activities (activities that depend on this activity). The *Block* rule ignores the change and does not modify *SuperNode*. The *Ask* rule defines that the system should ask a user to decide what to do at the moment of the change occurrence. The *Action* rule also

instructs that the ETL activity (*SuperNode*) should be modified accordingly to the change. Contrary to the *Propagate* rule, only input attributes of the activity should be modified. The evolution of the ETL process should stop on this activity and information about the change should not be passed to the next activities.

Table 1 presents all types of EDS structural changes and rules that can be set for them. *Inherit*, *Block*, and *Ask* rules work for every type of change in the same way. Contrary to this, the *Propagate* and *Action* rules are different for every type of change.

Table 1 Monitored structural changes and possible rules to define

Change type		Inherit	Propagate	Block	Ask	Action
Any change		✓	✓	✓	✓	✓
Collection	Add	✓	Join	✓	✓	Ignore
	Delete	✓	Delete	✓	✓	Replace
	Rename	✓	Rename	✓	✓	Map
	Split	✓	Delete	✓	✓	Merge
	Merge	✓	Add	✓	✓	Ignore
Collection element	Add	✓	Add	✓	✓	Ignore
	Delete	✓	Delete	✓	✓	Replace
	Rename	✓	Rename	✓	✓	Map
	Type	✓	Change type	✓	✓	Convert
	Length	✓	Change length	✓	✓	Cast

Every ETL activity represented by *SuperNode* can work in a different way. For example, it can be a simple SQL query, or it can just count duplicated elements. For a simple SQL query, a change like adding attribute may modify both input and output *Nodes*. However, for an activity that counts elements, a similar change may modify only the input *Nodes*. The output remains just as one numeric value. Activities based on SQL queries are similar and can be handled by rewriting the query. Contrary to this, activities that are based on ETL tool built-in functionality (i.e. fuzzy lookup) are more complex and each of them has its own parameters set that can be modified. Therefore, for every type of activity there must be a method for handling all types of changes. Since every ETL development environment can have a different set of available ETL activities and they can work in a different way, the handling methods are specific for every ETL development environment.

6 Related Work

The research and technological developments in the area of handling structural changes of EDSs in the DW architecture have mainly focused on managing changes in a DW. In this field, the five following approaches can be distinguished: (1) materialized view adaptation, (2) schema and data evolution, (3) temporal schema and data

extensions, (4) partial versioning of schema and data, and (5) the Multiversion Data Warehouse approach. Since they are not directly related to the topic of this paper, they will not be described here. An overview of research problems and approaches can be found in [7, 8].

Detecting structural changes in EDSs and propagating them into the ETL layer did not receive much attention from the research community. One of the first solution of this problem was Evolvable View Environment (EVE) presented in [5]. EVE is the environment that allows the evolution of an ETL process implemented by means of views. For every view it is possible to specify which elements of the views may change. It is possible to determine whether a particular attribute, both in the *select* and *where* clauses, can be omitted, or replaced by another attribute. Another possibility is that for every table, which is referred by a given view, a user can define whether this table can be omitted or replaced by another table.

The *E-ETL* versus EVE. *E-ETL* also employ a similar solution for handling missing elements (the *Replacer* algorithm). However, the *E-ETL* extends to this solution. *E-ETL* work with different ETL engines, whereas EVE works with ETL workflows developed as sequences of SQL queries. This difference implies that in *E-ETL* method for replacing missing elements can be applied not only for views, tables and their columns but also for ETL activities and their attributes.

Recent developments in the field of evolving ETL processes include a framework called *Hecataeus* [2, 3]. In *Hecataeus*, all ETL activities and EDSs are modeled as a graph whose nodes are relations, attributes, queries, conditions, views, functions, and ETL steps. Nodes are connected with edges that represent relationships between different nodes. The graph is annotated with rules that define the behavior of an ETL process in response to a certain EDS change event. In a response to an event, *Hecataeus* can either propagate the event, i.e. modify the graph according to a predefined policy, or prompt an administrator, or block the event propagation.

***E-ETL* versus *Hecataeus*.** The *E-ETL* framework, presented in this paper, is related to *Hecataeus*. However, *E-ETL* differs from *Hecataeus* with respect to:

- *E-ETL* has extended set of evolution rules;
- *E-ETL* has introduced new algorithms for repairing ETL process;
- *E-ETL* detects structural changes in EDSs either by means of schema triggers or by comparing two consecutive snapshots of EDS metadata (no information was provided how *Hecataeus* detects structural changes);
- *E-ETL* can be connected to any ETL engine and development environment that offers API, whereas *Hecataeus* needs a specific ETL engine that models ETL tasks by means of graphs;
- *E-ETL* support ETL workflows built of several complex operations (i.e. the operation of removing duplicates that may be available only in the external ETL tool), whereas *Hecataeus* work with ETL workflows developed as sequences of SQL queries;
- *E-ETL* can work with different types of EDS (i.e. data base, XML files, spreadsheet, record files), whereas *Hecataeus* supports only data bases as EDSs.

In [9] authors proposed a prototype system that can automatically detect changes in EDSs and propagate them into a DW. The prototype allows to define changes that are to be detected and associates with the changes actions executed in a DW. The main limitation of the prototype is that it does not allow ETL processes to evolve. Instead of that it focuses on propagating EDSs' changes into a DW. Moreover, the presented solution is restricted to only relational databases as EDSs. The next drawback of this prototype is a detection of changes which depends on triggers mechanism that can be not allowed to be installed in an EDS. Although, the *E-ETL* project is based on that developments, all mentioned shortcomings are not present in the *E-ETL* framework. Previous works on *E-ETL* were presented in [6].

7 Summary

This paper presents the *E-ETL* framework for detecting structural changes in EDSs and repairing an ETL process accordingly do detected changes. The framework repairs automatically an ETL process using evolution rules defined by a user. The *E-ETL* framework is also able to present to a user possible consequences of future changes (impact analyses). Currently we are implementing the presented framework. We are also preparing tests in an environment including structural changes that appeared in the real production DW systems, outlined in Section 5. Furthermore, we focus on developing a language for defining structural changes that are to be detected and propagated, and for repairing algorithms. *E-ETL* API is currently under development for communicating with Microsoft ETL engine, i.e., SQL Server Integration Services.

The approaches outlined in Section 6 handle structured changes in EDSs. However, as stressed in [14] even ordinary content (data) changes of an EDS may cause structural changes at a DW or changes to the structure of dimension data in a DW. Neither Hecataeus nor EVE nor [9] nor *E-ETL* supports handling appropriately such content changes. In future, we will work on handling such kinds of content changes at the ETL layer and on correctly propagating them into a DW.

References

1. Eder, J., Koncilia, C., Morzy, T.: The COMET Metamodel for Temporal Data Warehouses. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T. (eds.) CAiSE 2002. LNCS, vol. 2348, pp. 83–99. Springer, Heidelberg (2002)
2. Papastefanatos, G., Vassiliadis, P., Simitsis, A., Sellis, T., Vassiliou, Y.: Rule-Based Management of Schema Changes at ETL Sources. In: Grundspenkis, J., Kirikova, M., Manolopoulos, Y., Novickis, L. (eds.) ADBIS 2009. LNCS, vol. 5968, pp. 55–62. Springer, Heidelberg (2010)
3. Papastefanatos, G., Vassiliadis, P., Simitsis, A., Vassiliou, Y.: Policy-Regulated Management of ETL Evolution. *J. Data Semantics*, 147–177 (2009)

4. Rundensteiner, E.A., Koeller, A., Zhang, X.: Maintaining data warehouses over changing information sources. *Communications of the ACM* 43(6), 57–62 (2000)
5. Rundensteiner, E.A., Koeller, A., Zhang, X., Lee, A.J., Nica, A., Van Wyk, A., Lee, Y.: Evolvable View Environment (EVE): Non-Equivalent View Maintenance under Schema Changes. In: *Proc. of ACM Int. Conf. on Management of Data, SIGMOD*, pp. 553–555. ACM Press (1999)
6. Wojciechowski, A.: E-ETL: Framework For Managing Evolving ETL Processes. In: *Proc. of Ph.D. Students in Information and Knowledge Management Workshop (PIKM)*, pp. 59–66. ACM Press (2011)
7. Wojciechowski, A., Wrembel, R.: Research Problems of the ETL Technology. *Foundations of Computing and Decision Sciences* 35(5), 283–306 (2010)
8. Wrembel, R.: On handling the evolution of external data sources in a data warehouse architecture. In: Taniar, D., Chen, L. (eds.) *Data Mining and Database Technologies: Innovative Approaches*. IGI Group (2011)
9. Wrembel, R., Bębel, B.: The Framework for Detecting and Propagating Changes from Data Sources Structure into a Data Warehouse. *Foundations of Computing & Decision Sciences* 30(4), 361–372 (2005)

Author Index

- Appel, Ana Paula 305
Augustyn, Dariusz Rafał 3
Aydin, Secil 349
- Baziński, Bartosz 273
Bednář, Pavel 13
Bouju, Alain 187
Breß, Sebastian 27
Brzezicki, Michał 273
- Ceglarek, Dariusz 49
Chovanec, Peter 13
Correal, Dario 403
Cuzzocrea, Alfredo 59
- Deckert, Magdalena 69
Dembczyński, Krzysztof 79, 339
Djeddi, Warith Eddine 175
Dráždilová, Pavla 285
- Gajdoš, Petr 13
Gancarczyk, Joanna 373
Gaweł, Przemysław 79, 339
Geist, Ingolf 27
Grzegorowski, Marek 89
- Hammer, Barbara 141
Haniewicz, Konstanty 49
Horák, Zdeněk 317
- Janczak, Marcin 99
Jaskiewicz, Andrzej 79, 339
Jędrzejewski, Krzysztof 383
Jóźwiak, Ireneusz 425
- Kaczmarek, Krzysztof 37
Kajdanowicz, Tomasz 99
Kaleta, Mariusz 211
Kazienko, Przemysław 99
Khadir, Mohamed Tarek 175
Kotłowski, Wojciech 339
Krasuski, Adam 109
Krátký, Michal 13
Krawczyk, Bartosz 119
Kreński, Karol 109
Kubalík, Jiří 295
Kubiak, Marek 339
Kudělka, Miloš 317
Kunc, Petr 327
- Łazowy, Stanisław 109
- Malki, Jamal 187
Martinovič, Jan 285
Matoušek, Kamil 295
Mishra, Alok 349
Mishra, Deepti 349
Modliński, Piotr 221
- Nahorski, Zbigniew 231
Návrát, Pavol 395
Nečaský, Martin 295
Nguyen, Filip 327
Nienartowicz, Łukasz 415
- Ouzegane, Redouane 199
- Pałka, Piotr 231, 241
Pardel, Przemysław Wiktor 89
Paulovič, Aurel 395

- Pedraza-Garcia, Gilberto 403
Pereira, Adan Lucio 305
Pitner, Tomáš 327
- Quafafou, Mohamed 199
- Radvanský, Martin 131, 317
Radziszewska, Weronika 231
Ryžko, Dominik 251
Rzążewski, Paweł 37
- Schallehn, Eike 27
Schleif, Frank-Michael 141
Schoeneich, Radosław 241
Sklenář, Vladimír 131
Škrabálek, Jaroslav 327
Slaninová, Kateřina 285
Ślęzak, Dominik 109
Śliwiński, Tomasz 261
Snášel, Václav 131, 317
Sobolewski, Piotr 153
Stawicki, Sebastian 89
Stefanowski, Jerzy 69
Stencel, Krzysztof 89
- Susmaga, Robert 79, 339
Szczepanik, Michał 425
- Tari, Abdelkamel 199
- Vasilik, Dmitri 433
Vincent, Cécile 187
Vojtáš, Peter 295
- Wannous, Rouaa 187
Warczynski, Paweł 361
Wesołek, Przemysław 79, 339
Wojciechowski, Adam 361
Wojciechowski, Artur 441
Woźniak, Michał 153
Wróblewska, Anna 251
- Younsi, Zineb 199
- Zamorski, Maurycy 383
Zederowski, Sebastian 3
Zhu, Xibin 141
Zielniewicz, Piotr 79, 339
Ziemiński, Radosław Z. 163