

Barbara Catania • Tania Cerquitelli
Silvia Chiusano • Giovanna Guerrini
Mirko Kämpf • Alfons Kemper
Boris Novikov • Themis Palpanas
Jaroslav Pokorný • Athena Vakali *Editors*

New Trends in Databases and Information Systems

Selected Papers of the 17th East European Conference
on Advances in Databases and Information Systems
and Associated Satellite Events, Genoa, Italy,
September 1–4, 2013

Advances in Intelligent Systems and Computing

Volume 241

Series Editor

Janusz Kacprzyk, Warsaw, Poland

For further volumes:

<http://www.springer.com/series/11156>

Barbara Catania · Tania Cerquitelli
Silvia Chiusano · Giovanna Guerrini
Mirko Kämpf · Alfons Kemper
Boris Novikov · Themis Palpanas
Jaroslav Pokorný · Athena Vakali
Editors

New Trends in Databases and Information Systems

Selected Papers of the 17th East European
Conference on Advances in Databases and
Information Systems and Associated
Satellite Events, Genoa, Italy,
September 1–4, 2013

Editors

Barbara Catania
Dipartimento di Informatica
Bioingegneria, Robotica e
Ingegneria dei Sistemi
Università di Genova
Genova, Italy

Tania Cerquitelli
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, Italy

Silvia Chiusano
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, Italy

Giovanna Guerrini
Dipartimento di Informatica
Bioingegneria, Robotica e
Ingegneria dei Sistemi
Università di Genova
Genova, Italy

Mirko Kämpf
Cloudera, Inc.
California, USA

Alfons Kemper
Faculty of Informatics
Technische Universität München
Garching, Germany

Boris Novikov
Dept. of Analytical Information Systems
Saint Petersburg University
Saint Petersburg,
Russia

Themis Palpanas
Dipartimento di Ingegneria e Scienza
dell'Informazione
Università di Trento
Povo, TN, Italy

Jaroslav Pokorný
Department of Software Engineering
Faculty of Mathematics and Physics
Charles University, Praha
Czech Republic

Athena Vakali
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece

ISSN 2194-5357

ISBN 978-3-319-01862-1

DOI 10.1007/978-3-319-01863-8

Springer Cham Heidelberg New York Dordrecht London

ISSN 2194-5365 (electronic)

ISBN 978-3-319-01863-8 (eBook)

Library of Congress Control Number: 2013945417

© Springer International Publishing Switzerland 2014

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

This volume contains a selection of the papers presented at the 17th East-European Conference on Advances in Databases and Information Systems (ADBIS 2013) and the associated satellite events, held on September 1–4, 2013 in Genoa, Italy.

The ADBIS series of conferences aims at providing a forum for the dissemination of research accomplishments and to promote interaction and collaboration between the database and information system research communities from Central and East European countries and the rest of the world. The ADBIS conferences provide an international platform for the presentation of research on database theory, development of advanced DBMS technologies, and their advanced applications. ADBIS 2013 continued the ADBIS series held in St. Petersburg (1997), Poznan (1998), Maribor (1999), Prague (2000), Vilnius (2001), Bratislava (2002), Dresden (2003), Budapest (2004), Tallinn (2005), Thessaloniki (2006), Varna (2007), Pori (2008), Riga (2009), Novi Sad (2010), Vienna (2011), Poznań (2012). The programme of ADBIS 2013 includes keynotes, research papers, and five satellite events, consisting of a Big Data special session, three thematic workshops, and a Doctoral Consortium. The general idea behind each satellite event was to collect contributions from some subdomains of the broad research areas of databases and information systems, representing new trends in these two important areas.

This volume contains fourteen papers selected as short contributions to be presented at the ADBIS conference as well as papers contributed by all associated satellite events. An introductory chapter summarizes the main issues and contributions of all the events whose papers are included in this volume. Each of the satellite events complementing the main ADBIS conference had its own international program committee, whose members served as the reviewers of papers included in this volume. The volume is divided into 6 parts, one devoted to ADBIS 2013 short contributions and each other to a single satellite event.

The selected short papers span a wide spectrum of topics in the database field and related technologies, related to different types of data (spatio-temporal,

time-series, XML, workflow instance data), different management issues (querying, access methods, query processing, benchmarking, data analysis, mining), different types of architectures (including heterogeneous and distributed contexts, like P2P and MapReduce environments). Information system design and service oriented architecture specification are also addressed by the selected papers.

The ADBIS Special Session on Big Data: New Trends and Applications (BiDaTA 2013) aims at providing a forum for researchers, professionals, and practitioners in the industry sectors to discuss the research issues and share new ideas and techniques for big data management and analysis. Eight papers have been selected for presentation at BiDaTA 2013 and are included in this volume.

The Second International Workshop on GPUs in Databases (GID 2013) is devoted to all subjects related to utilization of Graphics Processing Units in database environments. The concept of using GPUs in databases is relatively young and has not yet received enough attention. The intention of the GID workshop is to provide a discussion forum for industrial and scientific communities. Presentation of practical and theoretical research creates chances for fruitful cooperation between the two communities. Four papers have been selected for presentation at GID 2013 and are included in this volume.

The Second International Workshop on Ontologies Meet Advanced Information Systems (OAIS 2013) seeks scientists, engineers, educators, industry people, policy makers, decision makers, and others to share their insight, vision, and understanding of the ontologies challenges in Advanced Information Systems. Six papers have been selected for presentation at OAIS 2013 and are included in this volume.

The First International Workshop on Social Business Intelligence: Integrating Social Content in Decision Making (SoBI 2013) aims at putting together for the first time researchers and practitioners coming from different areas related to Social Business Intelligence for sharing their findings and cross-fertilizing their research. Four papers have been selected for presentation at SoBI 2013 and are included in this volume, together with an invited paper on the workshop topic.

Last but not least, the *Doctoral Consortium* is a forum for Ph.D. students to present their research ideas, confront them with the scientific community, receive feedback from senior mentors, socialize and tie cooperation bounds. Besides ten poster presentations, three papers have been selected for presentation and are included in this volume. They cover three very different topics, all quite relevant for emerging applications in the database and information system field: spatial indexes, recommender systems, and concept drift.

We would like to thank everyone who contributed to the success of ADBIS 2013. We thank the authors, who submitted papers to the conference and the satellite events. We have also been dependent on many members of the community offering their time in organisational and reviewing roles - we are very grateful for the energy and professionalism they have exhibited. A special thank to the Program Committee members as well as to the external reviewers of the main conference and of each satellite event, for their support in evaluating the

submitted papers, ensuring the quality of the scientific program. Thanks also to all the colleagues involved in the conference organization, as well as to workshop organizers, for their work and effort without which assembling this volume would not have been possible. A special thank is deserved by the ADBIS Steering Committee and, in particular, its Chair, Leonid Kalinichenko, for their help and guidance. Special thanks are due to the publishing team at Springer, for their valuable assistance during the preparation of this manuscript. The conference would not have been possible without our sponsors and supporters: Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi, Università di Genova, Camera di Commercio di Genova, Coop Liguria, Comune di Genova, CTI Liguria. Last, but not least, we thank the participants of ADBIS 2013 for having made our work useful. Welcome to Genoa for the 2013 edition of the ADBIS conference!

July 9, 2013

Barbara Catania
Tania Cerquitelli
Silvia Chiusano
Giovanna Guerrini
Mirko Kämpf
Alfons Kemper
Boris Novikov
Themis Palpanas
Jaroslav Pokorný
Athena Vakali

ADBIS 2013 Conference Organization

General Chair

Barbara Catania

University of Genoa, Italy

Program Committee Co-chairs

Jaroslav Pokorný

Charles University in Prague, Czech Republic

Giovanna Guerrini

University of Genoa, Italy

Workshop Co-chairs

Themis Palpanas

University of Trento, Italy

Athena Vakali

Aristotle University of Thessaloniki, Greece

PhD Consortium Co-chairs

Alfons Kemper

Technical University of Munich, Germany

Boris Novikov

St. Petersburg University, Russia

ADBIS Steering Committee Chair

Leonid Kalinichenko

Russian Academy of Science, Russia

ADBIS Steering Committee

Paolo Atzeni, Italy

Andras Benczur, Hungary

Albertas Caplinskas, Lithuania

Barbara Catania, Italy

Johann Eder, Austria

Hele-Mai Haav, Estonia

Theo Haerder, Germany

Mirjana Ivanovic, Serbia

Hannu Jaakkola, Finland

Marite Kirikova, Latvia

Mikhail Kogalovsky, Russia	Yannis Manolopoulos, Greece
Rainer Manthey, Germany	Manuk Manukyan, Armenia
Joris Mihaeli, Israel	Tadeusz Morzy, Poland
Pavol Návrat, Slovakia	Boris Novikov, Russia
Mykola Nikitchenko, Ukraine	Jaroslav Pokorný, Czech Republic
Boris Rachev, Bulgaria	Bernhard Thalheim, Germany
Gottfried Vossen, Germany	Tatjana Welzer, Slovenia
Viacheslav Wolfengagen, Russia	Robert Wrembel, Poland
Ester Zumpano, Italy	

Organizing Committee

Publicity Chair

Marco Mesiti	University of Milan, Italy
--------------	----------------------------

Web Chair

Federico Cavaleri	University of Genoa, Italy
-------------------	----------------------------

Local Arrangement Chair

Paola Podestà	IMATI-CNR, Genoa, Italy
---------------	-------------------------

Local Organizing Committee

Alessandro Solimando	University of Genoa, Italy
Beyza Yaman	University of Genoa, Italy

Supporting Companies and Institutions

Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi
 Università di Genova
 Camera di Commercio di Genova
 Coop Liguria
 Comune di Genova
 CTI Liguria

ADBIS 2013 Program Committee

Suad Alagic	University of Southern Maine, USA
Manish Kumar Anand	Salesforce, USA
Andreas Behrend	University of Bonn, Germany
Ladjel Bellatreche	LIAS/ENSMA, France
Michela Bertolotto	University College Dublin, Ireland
Nicole Bidoit	LRI University Paris Sud 11, France
Maria Bielikova	Slovak University of Technology in Bratislava, Slovakia
Iovka Boneva	University Lille 1, France
Omar Boucelma	LSIS- CNRS, France
Stephane Bressan	National University of Singapore
Davide Buscaldi	University Paris Nord 13, France
Albertas Caplinskas	Vilnius University, Lithuania
Boris Chidlovskii	XRCE, France
Ricardo Ciferri	Federal University of São Carlos, Brazil
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria, Italy
Todd Eavis	Concordia University, Canada
Johann Eder	University of Klagenfurt, Austria
Alvaro A. A. Fernandes	The University of Manchester, UK
Pedro Furtado	University of Coimbra, Portugal
Johann Gamper	Free University of Bozen-Bolzano, Italy
Matjaz Gams	Jozef Stefan Institute, Slovenia
Anastasios Gounaris	Aristotle University of Thessaloniki, Greece
Goetz Graefe	HP Labs, USA
Adam Grzech	Wroclaw University of Technology, Poland
Hele-Mai Haav	Tallinn University of Technology, Estonia
Melanie Herschel	University Paris Sud 11, France
Theo Härder	Technical University of Kaiserslautern, Germany

Mirjana Ivanovic	University of Novi Sad, Serbia
Hannu Jaakkola	Tampere University of Technology, Finland
Leonid Kalinichenko	Russian Academy of Science, Russia
Alfons Kemper	Technical University of Munich, Germany
Maurice van Keulen	University of Twente, The Netherlands
Marite Kirikova	Riga Technical University, Latvia
Margita Kon-Popovska	Sts Cyril and Methodius University, Macedonia
Georgia Koutrika	HP Labs, USA
Stanislaw Kozielski	Silesian University of Technology, Poland
Jan Lindström	IBM Helsinki, Finland
Yannis Manolopoulos	Aristotle University of Thessaloniki, Greece
Rainer Manthey	University of Bonn, Germany
Giansalvatore Mecca	University of Basilicata, Italy
Marco Mesiti	University of Milan, Italy
Paolo Missier	Newcastle University, UK
Bernhard Mitschang	University of Stuttgart, Germany
Irena Mlynkova	Charles University in Prague, Czech Republic
Martin Nečaský	Charles University in Prague, Czech Republic
Anisoara Nica	SAP, Canada
Nikolaj Nikitchenko	Kiev State University, Ukraine
Boris Novikov	University of St Petersburg, Russia
Kjetil Nørvåg	Norwegian University of Science and Technology, Norway
Torben Bach Pedersen	Aalborg University, Denmark
Dana Petcu	West University of Timisoara, Romania
Evaggelia Pitoura	University of Ioannina, Greece
Elisa Quintarelli	Politecnico di Milano, Italy
Peter Revesz	University of Nebraska, USA
Stefano Rizzi	University of Bologna, Italy
Henryk Rybiński	Warsaw University of Technology, Poland
Ismael Sanz	University Jaume I, Spain
Kai-Uwe Sattler	Technical University of Ilmenau, Germany
Klaus-Dieter Schewe	Software Competence Center, Austria
Marc H. Scholl	University of Konstanz, Germany
Holger Schwarz	University of Stuttgart, Germany
Bela Stantic	Griffith University, Australia
Yannis Stavrakas	IMIS, Greece
Janis Stirna	Stockholm University, Sweden
Ernest Teniente	Technical University of Catalunya, Spain
Goce Trajcevski	Northwestern University, USA
Olegas Vasilecas	Vilnius Gediminas Technical University, Lithuania
Krishnamurthy Vidyasankar	Memorial University, Canada
Gottfried Vossen	University of Münster, Germany
Fan Wang	Microsoft, USA
Tatjana Welzer	University of Maribor, Slovenia

Robert Wrembel
Esteban Zimányi

Poznań University of Technology, Poland
Free University of Bruxelles, Belgium

Additional Reviewers

David Fekete
Enrique Flores
Stéphane Jean
Christian Koncilia
Tomáš Kramár

University of Münster, Germany
Technical University of Valencia, Spain
University of Poitiers, France
Panoratio Database Images, Inc., Germany
Slovak University of Technology in Bratislava,
Slovakia

Jens Lechtenböcker
Svetlana Mansmann
Mirjana Mazuran
Tudor Miu
Róbert Móra

University of Münster, Germany
University of Konstanz, Germany
Politecnico di Milano, Italy
NewCastle University, UK
Slovak University of Technology in Bratislava,
Slovakia

Solon Pissis
Emanuele Rabosio
Gianna Reggio
Alessandro Solimando
Justas Trinkunas

HITS, Germany
Politecnico di Milano, Italy
University of Genoa, Italy
University of Genoa, Italy
Vilnius Gediminas Technical University,
Lithuania

Dušan Zeleník

Slovak University of Technology in Bratislava,
Slovakia

BiDaTA 2013 – Special Session on Big Data: New Trends and Applications

Chairs

Tania Cerquitelli	Politecnico di Torino, Italy
Silvia Chiusano	Politecnico di Torino, Italy
Mirko Kämpf	Cloudera, Inc., Palo Alto, California, USA

Program Committee

Alexander Borusan	Technical University of Berlin, Germany
Andreas Both	Unister GmbH, Germany
Michelangelo Ceci	University of Bari, Italy
Byung-Gon Chun	Microsoft, USA
Paolo Garza	Politecnico di Torino, Italy
Ziyang Liu	NEC Laboratories America
Stefano Paraboschi	University of Bergamo, Italy
Domenico Saccà	University of Calabria, Italy
Claudio Sartori	University of Bologna, Italy
Marco Luca Sbodio	IBM Research, Ireland
Rusty Sears	Microsoft, USA
Andrea Tagarelli	University of Calabria, Italy
Riccardo Torlone	Roma Tre University, Italy
Filip Zavoral	Charles University in Prague, Czech Republic

Additional Reviewers

Alexander Alexandrov	Technical University Berlin, Germany
Giacomo Domeniconi	University of Bologna, Italy
Giovanni Ponti	ENEA-Portici Research Center, Italy

GID 2013 – The Second International Workshop on GPUs in Databases

Chairs

Witold Andrzejewski
Krzysztof Kaczmarek
Tobias Lauer

Poznań University of Technology, Poland
Warsaw University of Technology, Poland
Jedox AG, Germany

Program Committee

Amitava Datta

University of Western Australia,
Commonwealth of Australia

Artur 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

University of Warsaw, Poland

Paweł Wojciechowski

Poznań University of Technology, Poland

OAIS 2013 – The Second International Workshop on Ontologies Meet Advanced Information Systems

Chairs

Ladjel Bellatreche	LIAS/ENSMA, France
Yamine Ait Ameer	IRIT-ENSEIHT, France

Program Committee

Idir Ait-Sadoune	Supélec, Paris, France
Simitsis Alkis	HP, USA
Djamal Benslimane	LIRIS, Lyon, France
Mohand Boughanem	IRIT, Toulouse, France
Dickson K.W. Chiu	University of Hong Kong, China
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria, Italy
Faiez Gargouri	ISIMSF, Sfax, Tunisia
Daniela Grigori	Lamsade, Paris Dauphine University, France
Francesco Guerra	University of Modena and Reggio Emilia, Italy
Stéphane Jean	University of Poitiers, France
Selma Khouri	ESI, Algiers, Algeria
Haridimos Kondylakis	FORTH-ICS and University of Crete, Greece
Manolis Koubarakis	University of Athens, Greece
Mimoun Malki	Sidi Bel-Abbès University, Algeria
Brahim Medjahed	Michigan University, USA
Oscar Romero Moral	Technical University of Catalunya, Spain
Carlos Ordonez	Houston University USA
Fernando Silva Parreiras	FUMEC University, Brazil
Dimitris Plexousakis	Crete University, Greece
Chantal Reynaud	LRI, Paris, France
David Tariar	Monash University, Australia
Abdelkamel Tari	Béjaia University, Algeria
Farouk Toumani	LIMOS, Clermont Ferrand, France
Leandro Krug Wives	Federal University of Rio Grande do Sul, Brazil
Robert Wrembel	Poznań University of Technology, Poland
Boufaïda Zizette	Constantine University, Algeria

SoBI 2013 – The First International Workshop on Social Business Intelligence: Integrating Social Content in Decision Making

Chairs

Matteo Golfarelli

University of Bologna, Italy

Stefano Rizzi

University of Bologna, Italy

Program Committee

Alberto Abelló

Technical University of Catalunya, Spain

Marie-Aude Aufaure

École Centrale Paris, France

Rafael Berlanga Llavori

University Jaume I, Spain

Furio Camillo

University of Bologna, Italy

Jérôme Darmont

University of Lyon, France

Umesh Dayal

HP Labs, USA

Ronen Feldman

Hebrew University, Israel

Alfio Ferrara

University of Milan, Italy

Patrick Marcel

University of Tour, France

Jose Norberto Mazón

University of Alicante, Spain

Paul McNamee

Johns Hopkins University, USA

Alkis Simitsis

HP Labs, USA

Juan Carlos Trujillo

University of Alicante, Spain

Additional Reviewers

Mario Cataldi

École Centrale Paris, France

Enrico Gallinucci

University of Bologna, Italy

Contents

New Trends in Databases and Information Systems: Contributions from ADBIS 2013	1
<i>Yamine Ait Ameur, Witold Andrzejewski, Ladjel Bellatreche, Barbara Catania, Tania Cerquitelli, Silvia Chiusano, Matteo Golfarelli, Giovanna Guerrini, Krzysztof Kaczmarski, Mirko Kämpf, Alfons Kemper, Tobias Lauer, Boris Novikov, Themis Palpanas, Jaroslav Pokorný, Stefano Rizzi, Athena Vakali</i>	
Part I: ADBIS Short Contributions	
New Ontological Alignment System Based on a Non-monotonic Description Logic	17
<i>Ratiba Guebaili-Djider, Aicha Mokhtari, Farid Nouioua, Narhimene Boustia, Karima Akli Astouati</i>	
Spatiotemporal Co-occurrence Rules	27
<i>Karthik Ganesan Pillai, Rafal A. Angryk, Juan M. Banda, Tim Wylie, Michael A. Schuh</i>	
R⁺⁺-Tree: An Efficient Spatial Access Method for Highly Redundant Point Data	37
<i>Martin Šumák, Peter Gurský</i>	
Labeling Association Rule Clustering through a Genetic Algorithm Approach	45
<i>Renan de Padua, Veronica Oliveira de Carvalho, Adriane Beatriz de Souza Serapião</i>	
Time Series Queries Processing with GPU Support	53
<i>Piotr Przymus, Krzysztof Kaczmarski</i>	

Rule-Based Multi-dialect Infrastructure for Conceptual Problem Solving over Heterogeneous Distributed Information Resources	61
<i>Leonid Kalinichenko, Sergey Stupnikov, Alexey Vovchenko, Dmitry Kovalev</i>	
Distributed Processing of XPath Queries Using MapReduce	69
<i>Matthew Damigos, Manolis Gergatsoulis, Stathis Plitsos</i>	
A Query Language for Workflow Instance Data	79
<i>Philipp Baumgärtel, Johannes Tenschert, Richard Lenz</i>	
When Too Similar Is Bad: A Practical Example of the Solar Dynamics Observatory Content-Based Image-Retrieval System	87
<i>Juan M. Banda, Michael A. Schuh, Tim Wylie, Patrick McInerney, Rafal A. Angryk</i>	
Viable Systems Model Based Information Flows	97
<i>Marite Kirikova, Mara Pudane</i>	
On Materializing Paths for Faster Recursive Querying	105
<i>Aleksandra Boniewicz, Piotr Wiśniewski, Krzysztof Stencel</i>	
XSLTMark II – A Simple, Extensible and Portable XSLT Benchmark	113
<i>Viktor Mašiček, Irena Holubová (Mlýnková)</i>	
ReMoSSA: Reference Model for Specification of Self-adaptive Service-Oriented-Architecture	121
<i>Sihem Cherif, Raoudha Ben Djemaa, Ikram Amous</i>	
DSD: A DaaS Service Discovery Method in P2P Environments	129
<i>Riad Mokadem, Franck Morvan, Chirine Ghedira Guegan, Djamel Benslimane</i>	
Part II: Special Session on Big Data: New Trends and Applications	
Designing Parallel Relational Data Warehouses: A Global, Comprehensive Approach	141
<i>Soumia Benkrid, Ladjel Bellatreche, Alfredo Cuzzocrea</i>	
Big Data New Frontiers: Mining, Search and Management of Massive Repositories of Solar Image Data and Solar Events	151
<i>Juan M. Banda, Michael A. Schuh, Rafal A. Angryk, Karthik Ganesan Pillai, Patrick McInerney</i>	

Extraction, Sentiment Analysis and Visualization of Massive Public Messages	159
<i>Jacopo Farina, Mirjana Mazuran, Elisa Quintarelli</i>	
Desidoo, a Big-Data Application to Join the Online and Real-World Marketplaces	169
<i>Daniele Apiletti, Fabio Forno</i>	
GraphDB – Storing Large Graphs on Secondary Memory	177
<i>Lucas Fonseca Navarro, Ana Paula Appel, Estevam Rafael Hruschka Junior</i>	
Hadoop on a Low-Budget General Purpose HPC Cluster in Academia	187
<i>Paolo Garza, Paolo Margara, Nicolò Nepote, Luigi Grimaudo, Elio Piccolo</i>	
Discovering Contextual Association Rules in Relational Databases	193
<i>Elisa Quintarelli, Emanuele Rabosio</i>	
Challenges and Issues on Collecting and Analyzing Large Volumes of Network Data Measurements	203
<i>Enrico Masala, Antonio Servetti, Simone Basso, Juan Carlos De Martin</i>	
Part III: The Second International Workshop on GPUs in Databases	
GPU-Accelerated Query Selectivity Estimation Based on Data Clustering and Monte Carlo Integration Method Developed in CUDA Environment	215
<i>Dariusz Rafal Augustyn, Lukasz Warchal</i>	
Exploring the Design Space of a GPU-Aware Database Architecture	225
<i>Sebastian Breß, Max Heimel, Norbert Siegmund, Ladjel Bellatreche, Gunter Saake</i>	
Dynamic Compression Strategy for Time Series Database Using GPU	235
<i>Piotr Przymus, Krzysztof Kaczmarek</i>	
Online Document Clustering Using GPUs	245
<i>Benjamin E. Teitler, Jagan Sankaranarayanan, Hanan Samet, Marco D. Adelfio</i>	

Part IV: The Second International Workshop on Ontologies Meet Advanced Information Systems

Using the Semantics of Texts for Information Retrieval: A Concept- and Domain Relation-Based Approach	257
<i>Davide Buscaldi, Marie-Noëlle Bessagnet, Albert Royer, Christian Sallaberry</i>	
A Latent Semantic Indexing-Based Approach to Determine Similar Clusters in Large-scale Schema Matching	267
<i>Seham Moawed, Alsayed Algergawy, Amany Sarhan, Ali Eldosouky, Gunter Saake</i>	
Poss – <i>SRIOQ</i>(\mathcal{D}): Possibilistic Description Logic Extension toward an Uncertain Geographic Ontology	277
<i>Safia Bal Bourai, Aicha Mokhtari, Faiza Khellaf</i>	
Ontology-Based Context-Aware Social Networks	287
<i>Maha Maalej, Achraf Mtibaa, Faïez Gargouri</i>	
Diversity in a Semantic Recommender System	297
<i>Latifa Baba-Hamed, Magloire Namber</i>	
Ontology - Driven Observer Pattern	307
<i>Amrita Chaturvedi, Prabhakar T.V.</i>	

Part V: The First International Workshop on Social Business Intelligence: Integrating Social Content in Decision Making

Towards a Semantic Data Infrastructure for Social Business Intelligence	319
<i>Rafael Berlanga, María José Aramburu, Dolores M. Llidó, Lisette García-Moya</i>	
Subjective Business Polarization: Sentiment Analysis Meets Predictive Modeling	329
<i>Caterina Liberati, Furio Camillo</i>	
Sentiment Analysis and City Branding	339
<i>Roberto Grandi, Federico Neri</i>	
A Case Study for a Collaborative Business Environment in Real Estate	351
<i>Nicoletta Dessì, Gianfranco Garau</i>	

**OLAP on Information Networks: A New Framework for
Dealing with Bibliographic Data** 361
Wararat Jakawat, Cécile Favre, Sabine Loudcher

Part VI: Doctoral Consortium

Spatial Indexes for Simplicial and Cellular Meshes 373
Riccardo Fellegara

**Mathematical Methods of Tensor Factorization Applied to
Recommender Systems** 383
Giuseppe Ricci, Marco de Gemmis, Giovanni Semeraro

**Extended Dynamic Weighted Majority Using Diversity to
Handle Drifts** 389
Parneeta Sidhu, M.P.S. Bhatia

Author Index 397

Editors

Barbara Catania (*Contact editor*)

Dipartimento di Informatica,
Bioingegneria, Robotica e
Ingegneria dei Sistemi
Università di Genova
Via Dodecaneso 35
16146 Genova, Italy
barbara.catania@unige.it

Tania Cerquitelli

Dipartimento di Automatica e
Informatica
Politecnico di Torino
Corso Duca degli Abruzzi 24
10129 Torino, Italy
tania.cerquitelli@polito.it

Silvia Chiusano

Dipartimento di Automatica e
Informatica
Politecnico di Torino
Corso Duca degli Abruzzi 24
10129 Torino, Italy
silvia.chiusano@polito.it

Giovanna Guerrini

Dipartimento di Informatica,
Bioingegneria, Robotica e
Ingegneria dei Sistemi
Università di Genova
Via Dodecaneso 35
16146 Genova, Italy
giovanna.guerrini@unige.it

Mirko Kämpf

Cloudera, Inc.
220 Portage Ave
Palo Alto, CA 94306, USA
mirko.kaempf@cloudera.com

Alfons Kemper

Faculty of Informatics
Technische Universität München
Boltzmannstr. 3
85748 Garching, Germany
kemper@in.tum.de

Boris Novikov

Dept. of Analytical Information
Systems
Saint Petersburg University
Universitetsky prosp. 28
198504 Saint Petersburg, Russia
b.novikov@spbu.ru

Themis Palpanas

Dipartimento di Ingegneria e Scienza
dell'Informazione
Università di Trento
Via Sommarive 14
38123 Povo, TN, Italy
themis@disi.unitn.eu

Jaroslav Pokorný

Department of Software Engineering
Faculty of Mathematics and Physics
Charles University

Malostranské nám. 25
118 00 Praha 1, Czech Republic
pokorny@ksi.mff.cuni.cz

Athena Vakali

Department of Informatics
Aristotle University of Thessaloniki
AUTH campus
54124 Thessaloniki, Greece
avakali@csd.auth.gr

New Trends in Databases and Information Systems: Contributions from ADBIS 2013

Yamine Ait Ameer¹, Witold Andrzejewski², Ladjel Bellatreche³,
Barbara Catania⁴, Tania Cerquitelli⁵, Silvia Chiusano⁵, Matteo Golfarelli⁶,
Giovanna Guerrini⁴, Krzysztof Kaczmarski⁷, Mirko Kämpf⁸, Alfons Kemper⁹,
Tobias Lauer¹⁰, Boris Novikov¹¹, Themis Palpanas¹², Jaroslav Pokorný¹³,
Stefano Rizzi⁶, and Athena Vakali¹⁴

¹ IRIT-ENSEIHT, France

² Poznan University of Technology, Poland

³ LIAS/ENSMA, France

⁴ University of Genoa, Italy

⁵ Politecnico di Torino, Italy

⁶ University of Bologna, Italy

⁷ Warsaw University of Technology, Poland

⁸ Cloudera, Inc., Palo Alto, California, USA

⁹ Technical University of Munich, Germany

¹⁰ Jedox AG, Germany

¹¹ Saint Petersburg University, Russia

¹² University of Trento, Italy

¹³ Charles University in Prague, Czech Republic

¹⁴ Aristotle University of Thessaloniki, Greece

Abstract. Research on database and information system technologies has been rapidly evolving over the last few years. Advances concern either new data types, new management issues, and new kind of architectures and systems. The 17th East-European Conference on Advances in Databases and Information Systems (ADBIS 2013), held on September 1–4, 2013 in Genova, Italy, and associated satellite events aimed at covering some emerging issues concerning such new trends in database and information system research. The aim of this paper is to present such events, their motivations and topics of interest, as well as briefly outline the papers selected for presentations. The selected papers will then be included in the remainder of this volume.

1 Introduction

The East-European Conference on Advances in Databases and Information Systems (ADBIS) aims at providing a forum for the dissemination of research accomplishments and to promote interaction and collaboration between the database and information system research communities from Central and East European countries and the rest of the world. The ADBIS conferences provide an international platform for the presentation of research on database theory, development of advanced DBMS technologies, and their advanced applications. ADBIS 2013

continued the ADBIS series held in St. Petersburg (1997), Poznan (1998), Maribor (1999), Prague (2000), Vilnius (2001), Bratislava (2002), Dresden (2003), Budapest (2004), Tallinn (2005), Thessaloniki (2006), Varna (2007), Pori (2008), Riga (2009), Novi Sad (2010), Vienna (2011), Poznań (2012).

The programme of the 17th ADBIS conference, held on September 1-4, 2013 in Genoa, Italy, includes keynotes, research papers, and five satellite events. In 2013, satellite events include for the first time a special session on Big Data Management, with special emphasis on industrial applications, three thematic workshops, and the traditional Doctoral Consortium, for presentation of interesting PhD student work. While papers accepted at the ADBIS main conference span a wide spectrum of topics in the field of databases and information systems, ranging from semantic data management and similarity search, to spatio-temporal and social network data, data mining and data warehousing, data management on novel architectures (GPU, parallel DBMS, cloud and MapReduce environments), the general idea behind each satellite event 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. More precisely, the following satellite events have been organized:

- Special Session on Big Data: New Trends and Applications (BiDaTA 2013).
- The Second International Workshop on GPUs in Databases (GID 2013).
- The Second International Workshop on Ontologies Meet Advanced Information Systems (OAIS 2013).
- The First International Workshop on Social Business Intelligence: Integrating Social Content in Decision Making (SoBI 2013).
- Doctoral Consortium.

The main ADBIS conference as well as each of the satellite events had its own international program committee, whose members served as the reviewers of papers included in this volume.

This volume contains papers selected as short contributions to be presented at the ADBIS 2013 main conference as well as papers contributed by all satellite events listed above. In the following, for each event, we present its main motivations and topics of interest and we briefly outline the papers selected for presentations. The selected papers will then be included in the remainder of this volume. Some acknowledgements from the organizers are finally provided.

2 ADBIS Selected Short Contributions

Introduction. The ADBIS main conference was chaired by Giovanna Guerrini (University of Genoa, Italy) and Jaroslav Pokorný (Charles University in Prague, Czech Republic). The main conference attracted 92 paper submissions from 43 different countries representing all the continents. All papers were evaluated by at least three reviewers. As a result of a rigorous reviewing process, besides 26 papers selected as full contributions and published in the LNCS series, 14 papers were selected as short contributions and included in this volume. The Program

Committee was composed of 73 members, 16 additional reviewers further supported the review workload.

Selected papers. The selected 14 short papers span a wide spectrum of topics in the database field and related technologies.

Papers consider a wide variety of data, ranging from spatio-temporal to XML and workflow instance data, from the points of view of querying, access methods, query processing, and benchmarking. Specifically, the paper *R⁺⁺-tree: An Efficient Spatial Access Method for Highly Redundant Point Data* (Martin Šumák and Peter Gurský) proposes a spatial index structure defined as a variation on R⁺-trees offering even better search efficiency than R*-tree when highly redundant point data are considered. The paper *A Query Language for Workflow Instance Data* (Philipp Baumgärtel, Johannes Tenschert, and Richard Lenz) proposes a query language to aggregate and query workflow instance data, motivated by an application in a simulation system to be applied to the clinical domain, with the aim of supporting domain experts in analyzing simulation input and output. Efficient query processing on workflow definitions and instance data in RDF is also investigated. The paper *On Materializing Paths for Faster Recursive Querying* (Aleksandra Boniewicz, Piotr Wiśniewski, and Krzysztof Stencel) addresses the problem of efficient implementation of recursive rules and proposes the use of redundant data structures to answer recursive queries, investigating as well the overhead imposed by the synchronization of such structures upon updates. The paper *XSLTMark II - a Simple, Extensible and Portable XSLT Benchmark* (Viktor Mašiček and Irena Holubová) proposes a benchmark for XSLT, created on the basis of the analysis of real-world XSLT scripts. The benchmark allows one to generate test cases from templates of tests, run tests, produce XML reports, transform reports into HTML format and test different XSLT processors.

Query processing is addressed as well in the “Big Data” context, considering large XML data and time series data, and paradigms such as MapReduce as well as GPUs. Specifically, the paper *Distributed Processing of XPath Queries using MapReduce* (Matthew Damigos, Manolis Gergatsoulis, and Stathis Plitso) proposes a MapReduce algorithm for evaluating XPath queries over large XML data stored in a distributed manner. The algorithm is based on a query decomposition which computes all expected answers in one MapReduce step. The paper *Time Series Queries Processing with GPU support* (Piotr Przymus and Krzysztof Kaczmarek), by contrast, copes with time series data. It presents a prototype query engine based on GPU and NoSQL databases plus a new model of data storage using lightweight compression.

Besides data management and querying, data analysis and mining is addressed as well. Specifically, the paper *Labeling Association Rule Clustering through a Genetic Algorithm Approach* (Renan de Padua, Veronica Oliveira de Carvalho, and Adriane Beatriz de Souza Serapião) focuses on the post-processing of association rules, and specifically on their clustering. When an association rule set is clustered, an improved presentation of the mined patterns is shown to the

user, provided that good labels are assigned to the groups, in order to guide the user during the exploration process. The paper *Spatiotemporal Co-occurrence Rules* (Karthik Ganesan Pillai, Rafal A. Angryk, Juan M. Banda, Tim Wylie, and Michael A. Schuh) presents a general framework to discover spatiotemporal co-occurrence rules for continuously evolving spatiotemporal events that have extended spatial representations. The discovery of such rules is an important problem in many application domains such as weather monitoring and solar physics. The paper *When Too Similar is Bad: A Practical Example of the Solar Dynamics Observatory Content-Based Image-Retrieval System* (Juan M. Banda, Michael A. Schuh, Tim Wylie, Patrick McInerney, and Rafal A. Angryk) addresses an important image data mining and information retrieval issue: finding similar images, which correspond to temporal neighbors capturing the same event instance, i.e., similar solar events in the context of the reference Solar Dynamics Observatory.

Selected papers target as well heterogeneous and distributed contexts, ranging from semantic data management (ontology alignment) to querying heterogeneous information sources and service discovery in P2P architectures. The paper *New Ontological Alignment System based on a Non Monotonic Description Logic* (Ratiba Guebaili Djider, Aicha Mokhtari, Farid Nouioua, Narhimene Boustia, and Karima Akli Astouati) considers the use of a non monotonic description logic, with an algebraic semantics, capable of assuring a maximum of expressiveness in the definition of ontologies concepts and relationships by taking into account the normal context aspect and the exception one. On this basis, ontology alignment and related structural similarity measures are then investigated. The paper *DSD: a DaaS Service Discovery Method in P2P Environments* (Riad Mokadem, Franck Morvan, Chirine Ghedira Guegan, and Djamel Benslimane) deals with service discovery in Data as a Service distributed P2P environments. The proposed discovery method does not impose any topology on the graph formed by domain ontologies and mapping links. Peers, using a common domain ontology, are grouped in a Virtual Organization and connected in a Distributed Hash Table. The paper *Rule-based Multi-dialect Infrastructure for Conceptual Problem Solving over Heterogeneous Distributed Information Resources* (Leonid Kalinichenko, Sergey Stupnikov, Alexey Vovchenko, and Dmitry Kovalev) proposes an approach for applying a combination of semantically different rule-based languages for interoperable conceptual programming over various rule-based systems relying on the logic program transformation technique recommended by the W3C Rule Interchange Format (RIF). The approach is combined with heterogeneous database integration by applying semantic rule mediation.

Finally, also information system design and service oriented architecture specification are addressed by selected papers. The paper *Viable Systems Model Based Information Flows* (Marite Kirikova and Mara Pudane) deals with information system engineering and, specifically, with how to ensure that all essential information flows are properly identified and supported. A Viable Systems Model (VSM) is used as a basis for the identification of a set of information flows, which should be present in VSM complying enterprises. The paper *ReMoSSA: Reference Model for*

Specification of Self Adaptive Service-Oriented-Architecture (Sihem Cherif, Raoudha Ben Djemaa, and Ikram Amous) proposes a reference model for specifying self-adaptive Service-Based Applications. The proposed model integrates self-adaptation mechanisms and strategies to provide autonomic and adaptable service, thus reducing maintenance costs and efforts.

3 BiDaTA 2013 – Special Session on Big Data: New Trends and Applications

Introduction. The Special Session on Big Data - New Trends and Applications (BiDaTA 2013) has been organized by Tania Cerquitelli (Politecnico di Torino, Italy), Silvia Chiusano (Politecnico di Torino, Italy), and Mirko Kämpf (Cloudera, Inc., Palo Alto, California, USA).

Large volumes of data (Big Data) are being produced by various modern applications at an ever increasing rate. These applications range from wireless sensor networks (e.g., climate/weather monitoring, intelligent mobility, water metering) to social networks and e-commerce applications. Innovative data models, algorithms, and architectures have to be designed to deal with the “Big Data four V-dimensions”, namely Volume, Velocity, Variety, and Veracity. These new paradigms of software and hardware design should efficiently store, manage, and analyze such huge data volumes, providing the necessary scalability and flexibility for novel big data analytics applications. These challenges have been attracting great attention from both academia and industry. The BiDaTA session aims at providing a forum for researchers, professionals, and practitioners in the industry sectors to discuss the research issues and share new ideas and techniques for big data management and analysis. Topics of interest for this session range from big data models, algorithms, and architectures, to cloud computing techniques for big data, and big data search and mining in different application domains. BiDaTA welcome research papers, application papers, and papers on experience reports on various aspects of big data.

The BiDaTA Program Committee was composed of 14 members. The reviewing process was also supported by 3 additional reviewers.

Keynote presentations. In the era of big data, new software design paradigms are needed to provide the necessary scalability and flexibility in developing novel big data analytics applications. The Apache Hadoop software library is a widely used open-source framework supporting reliable, scalable, and distributed software running across clusters of computers. Based on the work done by Google in the late 1990s and the early 2000s, Hadoop is continuously evolving to meet new trends and emerging needs in processing large data volumes in various application domains (e.g., social networks and medical domain). The BiDaTA session includes two keynote presentations held by Lars George (Director EMEA Services at Cloudera) and Carlo Curino (Senior Scientist at Microsoft, USA) related to Apache Hadoop. While the former discusses the main evolutions of the Hadoop framework, the latter presents his experience on how to get involved in

the Hadoop open-source project. The two keynotes are briefly described in the following, together with a short biography of the speakers.

“Hadoop is Dead, Long Live Hadoop!”, by *Lars George*. Hadoop has made its way from a batch-oriented storage and processing framework to a fully fledged, enterprise compatible ecosystem that harbours many additional projects that are needed to move data in and out, as well as to process it timely. Following the Google-led timeline of additions to this framework plots a clear way ahead into less batch-oriented workloads, like quick exploration and mining of more specific data sets – often requiring its own **structured** file format. Algorithms less amicable to the MapReduce framework find their way into the ecosystem by means of more generic resource management frameworks, such as YARN. This talk addresses the current status of the Hadoop platform, yet also raises questions and ideas on where Hadoop as an ecosystem is growing into. Hadoop has become more than what it was originally, it is a new system with huge potential for research projects as a platform as well as a Petri dish for new developments within itself. Long live Hadoop!

Lars George has been involved with HBase since 2007, and became a full HBase committer in 2009. He has spoken at many conferences and Hadoop User Group meetings, such as ApacheCon, FOSDEM, QCon, JAX, or Hadoop World and Summit. He also started the Munich OpenHUG meetings. Lars now works for Cloudera, as the Director EMEA Services, managing a team of Hadoop solutions architects in and around Europe. He is also the author of O’Reilly’s “HBase - The Definitive Guide”.

“Big-Data Services in the Azure Cloud”, by *Carlo Curino*. The talk presents the evolution of Hadoop from a MapReduce-only framework towards a fully general resource management framework (YARN) enabling arbitrary data-intensive programming models to co-exist. The experience acquired on supporting work-preserving preemption and how to improve the YARN resource scheduling aspects to increase cluster utilization is also discussed. Furthermore, a path towards a next-generation, highly multi-tenant Hadoop cloud offering, and how YARN can be leveraged for research purposes, is highlighted.

Carlo Curino received a PhD from Politecnico di Milano, and spent two years as Post Doc Associate at CSAIL MIT leading the relational cloud project. He worked at Yahoo! Research as Research Scientist focusing on mobile/cloud platforms and entity deduplication at scale. Carlo is currently a Senior Scientist at Microsoft in the recently formed Cloud and Information Services Lab (CISL) where he is working on big-data platforms and cloud computing.

Selected papers. The special session is composed of 8 papers discussing different interesting research issues, application domains, and experience reports on big data management and analysis. Specifically, the session contains 4 research papers, 2 application papers, and 2 experience reports. Due to the various topics covered by the papers, in the following we interleave the presentation of research and application papers, as well as experience reports.

Parallel Relational Data Warehouses (PRDW) have been proposed as a scalable platform for storing, processing and analyzing large data volumes. The research paper *Designing Parallel Relational Data Warehouses: a Global, Comprehensive Approach* (Soumia Benkrid, Ladjel Bellatreche, and Alfredo Cuzocrea) addresses the data replication issues in designing PRDWs. The authors present a redundant allocation algorithm, based on the fuzzy k-means clustering algorithm, to design shared-nothing PRDWs.

Solar physics is an emerging big data research domain due to the massive amounts of data generated daily. The application paper *Big Data New Frontiers: Mining, Search and Management of Massive Repositories of Solar Image Data and Solar Events* (Juan M. Banda, Michael A. Schuh, Rafal A. Angryk, Karthik Ganesan Pillai, and Patrick McInerney) describes an interesting experience to efficiently manage, search, and mine large collections of solar image data and solar events. Methodologies and future directions for big data processing in solar physics are discussed.

Nowadays, communication technologies allow users to exchange huge amount of messages that, when properly analysed, can provide insights into user opinions. The research paper *Extraction, Sentiment Analysis and Visualization of Massive Public Messages* (Jacopo Farina, Mirjana Mazuran, and Elisa Quintarelli) proposes a framework, running in a distributed environment, for the extraction, sentiment analysis, and visualization of a large amount of public messages from diverse sources (e.g., social networks).

The experience report *Desidoo, a Big-Data Application to Join the Online and Real-World Marketplaces* (Daniele Apiletti and Fabio Forno) discusses the industry experience to realize an innovative big-data marketplace service running in the cloud that couples virtual and physical shops. Many challenges and issues are discussed, ranging from dealing with heterogenous data to scaling the proposed platform.

Recently, the volume of complex network data has increased exponentially, while most mining algorithms assume that the network fits in primary memory. Consequently, efficiently storing and retrieving big network data is a great challenge. The research paper *GraphDB - Storing large graphs on secondary memory* (Lucas Fonseca Navarro, Ana Paula Appel, and Estevam Rafael Hruschka Junior) presents a novel persistent data structure to store, access, and query large complex networks.

Using High Performance Computing (HPC) infrastructures for data intensive application is an important issue in different application contexts. The experience report *Hadoop on a Low-Budget General Purpose HPC Cluster in Academia* (Paolo Garza, Paolo Margara, Nicolò Nepote, Luigi Grimaudo, and Elio Piccolo) describes the experience made in integrating Hadoop in an academic HPC cluster to jointly provide all available services based on MPI applications together with the new ones based on Hadoop.

Context-aware systems can be adopted to mine only the relevant knowledge from large data collections. These systems exploit the information on the user context to tailor the application behaviours to her needs. The research paper

Discovering Contextual Association Rules in Relational Databases (Elisa Quintarelli and Emanuele Rabosio) proposes a novel algorithm to efficiently mine contextual association rules in relational databases.

Finally, the application paper *Challenges and Issues on Collecting and Analyzing Large Volumes of Network Data Measurements* (Enrico Masala, Antonio Servetti, Simone Basso, and Juan Carlos De Martin) presents the open-source Neubot project collecting various network data measurements to analyze the performance of end-users' Internet connections. The authors discuss issues to efficiently query and analyze in real time the potentially large amount of collected data.

4 GID 2013 – The Second International Workshop on GPUs in Databases

Introduction. The Second International Workshop on GPUs in Databases (GID 2013) was organized by Witold Andrzejewski (Poznan University of Technology, Poland), Krzysztof Kaczmarski (Warsaw University of Technology, Poland), and Tobias Lauer (Jedox AG, Germany). GID is devoted to all subjects related to utilization of Graphics Processing Units in database environments. The concept of using GPUs in databases is relatively young and has not yet received enough attention. The intention of the GID workshop is to provide a discussion forum for industrial and scientific communities. Presentation of practical and theoretical research creates chances for fruitful cooperation between the two communities. The 2013 event is already the second edition of the workshop (the previous one was organized with ADBIS 2012 conference). The GID 2013 Program Committee was composed of 7 members.

Selected papers. Similarly to the previous edition, 4 interesting presentations were selected.

The paper *GPU-Accelerated Query Selectivity Estimation based on Data Clustering and Monte Carlo Integration Method developed in CUDA Environment* (Dariusz Rafal Augustyn and Lukasz Warchal) tackles the problem of utilizing GPUs for accurate and fast computation of query selectivity estimation based on space efficient data distribution representations.

The paper *Exploring the Design Space of a GPU-aware Database Architecture* (Sebastian Breß, Max Heimel, Norbert Siegmund, Ladjel Bellatreche, and Gunter Saake) introduces a survey of many approaches for utilizing GPUs in databases. Based on this survey, key properties, important trade-offs and typical challenges of using GPUs in database environments are identified, and open research problems are formulated.

The paper *Dynamic Compression Strategy for Time Series Database using GPU* (Piotr Przymus and Krzysztof Kaczmarski) shows a very fast GPU accelerated lossless compression algorithm for time series databases.

Finally, the paper *Online Document Clustering Using GPUs* (Benjamin E. Teitler, Jagan Sankaranarayanan, Hanan Samet, and Marco D. Adelfio) tackles

the problem of clustering multiple documents in parallel by utilizing efficient parallel processing capabilities of GPUs.

5 OAIS 2013 – The Second International Workshop on Ontologies Meet Advanced Information Systems

Introduction. The Second International Workshop on Ontologies Meet Advanced Information Systems (OAIS 2013) was chaired by Ladjel Bellatreche (LIAS/ENSMA, France) and Yamine Ait Ameur (IRIT-ENSEIHT, France).

Information Systems are record sensitive and rely on crucial data to support day-to-day company applications and decision making processes. Therefore, these systems often contain most of company products and process knowledge. Unfortunately, this knowledge is implicitly encoded within the semantics of the modelling languages used by the companies. The explicit semantics is usually not recorded in such models of information systems. References to ontologies could be considered as an added value for handling the explicit semantics carried by the concepts, data and instances of models. Thus, developing new user interfaces or reconciling data and/or models with external ones often require some kind of reverse engineering processes for making data semantic explicit.

Nowadays, ontologies are used for making explicit the meaning of information in several research and application domains. Ontologies are now used in a large spectrum of fields such as: Semantic Web, information integration, database design, e-Business, data warehousing, data mining, system interoperability, formal verification. They are also used to provide information systems with user knowledge-level interfaces. Over the last five years, a number of interactions between ontologies and information systems have emerged. New methods have been proposed to embed within databases both ontologies and data, defining new ontology-based database systems. New languages were developed in order to facilitate exchange of both ontology and data. Other languages dedicated to query data at the ontological level were proposed (e.g., RQL, SOQA-QL, or OntoQL). In some domains, like social networks, recommender systems, information retrieval, geographic information systems, concurrent engineering, etc. the ontologies are used to define world wide exchange consortiums for identifying relevant information, recommending them, providing semantic indexes, matching schemas of heterogeneous information sources, etc.

All these motivations led to the organization of OAIS 2013. This event intentionally sought scientists, engineers, educators, industry people, policy makers, decision makers, and others to share their insight, vision, and understanding of the ontologies challenges in Advanced Information Systems. The OAIS Program Committee was composed of 27 members.

Selected papers. We accepted 6 papers from various countries all over the world (Algeria, France, Germany, India, and Tunisia). The paper *Using the Semantics of Texts for Information Retrieval: a Concept- and Domain Relation-based Approach* (Davide Buscaldi, Marie-Noëlle Bessagnet, Albert Royer, and

Christian Sallaberry) presents a method for calculating conceptual similarity. The used information retrieval strategy is based on exploring an ontology and domain relations between concepts marked by verbal forms. Experiments executed using the implemented system show that using ontologies improves recall with respect to a classic Information Retrieval system. When also domain relations are considered, precision is also improved.

The paper *A Latent Semantic Indexing-based Approach to Determine Similar Clusters in Large-Scale Schema Matching* (Seham Moawad, Alsayed Algergawy, Amany Sarhan, Ali Eldosouky, and Gunter Saake) deals with the identification of semantic correspondences across shared-data applications, such as data integration, and presents a new clustering-based approach, using Latent Semantic Indexing for retrieving the conceptual meaning between clusters.

The paper *Poss – SROIQ(D): Possibilistic Description Logic Extension Toward an Uncertain Geographic Ontology* (Safia Bourai, Aicha Mokhtari, and Faiza Khellaf) presents a possibilistic extension of Description Logic as a solution to handle uncertainty and to deal with inconsistency in geographical applications.

The paper *Ontology-based Context-Aware Social Networks* (Maha Maalej, Achraf Mtibaa, and Faiez Gargouri), after presenting a state-of-the-art survey about knowledge extraction using ontologies in social networks, proposes an approach which combines context-awareness and ontology usage in mobile platforms, with the aim of assisting a mobile user in retrieving her/his information from a social network.

The paper *Diversity in a Semantic Recommender System* (Latifa Baba-Hamed and Magloire Namber) introduces the notion of diversity in recommender systems, with the aim of developing algorithms to provide the user with not only all the most relevant contents, but also the most diversified.

Finally, the paper *Ontologydriven Observer Pattern* (Amrita Chaturvedi and Prabhakar T.V.) proposes an ontology driven observer pattern which mitigates the drawbacks arising in GoF observer patterns and also those which occur in the general usage of patterns.

6 SoBI 2013 – The First International Workshop on Social Business Intelligence: Integrating Social Content in Decision Making

Introduction. The First International Workshop on Social Business Intelligence: Integrating Social Content in Decision Making (SoBI 2013) was organized by Matteo Golfarelli (University of Bologna, Italy) and Stefano Rizzi (University of Bologna, Italy)

Social Business Intelligence is the discipline of effectively and efficiently combining corporate data with social data to let decision-makers effectively analyze and improve their business based on the trends and moods perceived from the environment. As in traditional Business Intelligence, the goal is to enable

powerful and flexible analyses for users with a limited expertise in databases and ICT.

Social Business Intelligence is at the cross-road of several areas in Computer Science such as Database Systems, Information Retrieval, Data Mining, and Natural Language Processing. Though the ongoing research in these fields has made available results and technologies for Social Business Intelligence, an overall view of the related problems and solutions is still missing. The goal of SoBI 2013 was to put together for the first time researchers and practitioners coming from different areas related to Social Business Intelligence for sharing their findings and cross-fertilizing their research.

The SoBI 2013 Program Committee included 13 members. The reviewing process was also supported by 2 additional reviewers. They carefully revised the papers submitted to SoBI as well as the papers initially submitted to the Second International Workshop on Social Data Processing (SDP 2013), organized by Jaroslav Pokorný (Charles University in Prague, Czech Republic), Katarzyna Wegrzyn-Wolska (ESIGETEL, France), and Vaclav Snasel (VSB - Technical University of Ostrava, Czech Republic), which was canceled due to the limited number of submitted papers.

Keynote presentations. The SoBI program included a keynote presentation, whose related paper is contained in this volume. The invited talk *Towards a Semantic Data Infrastructure for Social Business Intelligence*, given by Rafael Berlanga LLavori, aims at introducing the new challenges arising when attempting to integrate traditional corporate data and external sentiment data, to devise potential solutions for the near future, and to propose a semantic data infrastructure aimed at providing new opportunities for integrating traditional and social Business Intelligence.

Rafael Berlanga Llavori is associate professor of Computer Science at Universitat Jaume I, Spain, and the leader of the TKBG research group. He received the BS degree from Universidad de Valencia in Physics, and the PhD degree in Computer Science in 1996 from the same university. In the past, his research was focused on temporal reasoning and planning in AI. His current research interests include knowledge bases, information retrieval and the semantic web. He has directed several research projects and has published in several journals and international conferences in the above areas.

Selected papers. The SoBI program includes 4 research presentations. The paper *Subjective Business Polarization: Sentiment Analysis Meets Predictive Modeling* (Furio Camillo and Caterina Liberati) focuses on sentiment analysis, shows how a probabilistic Kernel classifier can be employed to get the rule of discrimination for automatically assigning a polarity to social content out of a manually labeled training set, and presents the results of a real case study related to a world-wide brand of beauty products.

The paper *Sentiment Analysis and City Branding* (Roberto Grandi and Federico Neri) aims at illustrating the potential of sentiment analysis. This is done

by presenting a case study that applies opinion mining to city branding aimed at showing what trends it can —and cannot— highlight.

The paper *A Case Study for a Collaborative Business Environment in Real Estate* (Nicoletta Dessì and Gianfranco Garau) falls in the area of collaborative decision support. The idea is to inject social perspective into a Spatial Decision Support System: decision makers are organized in a social structure that includes citizens, companies, and organizations and interact using a social network.

The paper *OLAP on Information Networks: a New Framework for Dealing with Bibliographic Data* (Wararat Jakawat, Cécile Favre, and Sabine Loudcher) discusses the main challenges arising when combining information networks, OLAP, and data mining technologies with specific reference to bibliographic data. The main idea is to be able to analyze these data and their dynamics adopting different points of view.

7 Doctoral Consortium

Introduction. The Doctoral Consortium (DC) is a forum for Ph.D. students to present their research ideas, confront them with the scientific community, receive feedback from senior mentors, socialize and tie cooperation bounds. Students receive support and inspiration from their peers, and they enjoy the opportunity to discuss their research and career objectives with senior members of the community from outside their institution. Students present and discuss their research directions in the context of an established international conference outside of their usual university environment. The chairs of the Ph.D. Consortium, responsible for selecting the papers from this category, were Alfons Kemper (Techn. Univ. Muenchen, Munich, Germany) and Boris Novikov (St-Petersburg University, Russia). The DC sessions were scheduled in parallel with workshop sessions affiliated with the ADBIS 2013 conference. Each participant had an opportunity to present her/his research, followed by discussion with and comments by senior researchers. In addition, a poster session was held during the main conference. Each participant of the Doctoral Consortium was invited to present her/his research during the poster session, thus increasing the exposure of the research and engaging in discussions with senior members of the research community.

Selected papers. We sought Ph.D. student participants who are either advanced and have determined the direction of their thesis research with some preliminary results already obtained or Ph.D. student participants who are in the early stages of their dissertation year. The Doctoral Consortium hosted 10 presentations of which 3 were accepted as advanced PhD projects with a paper publication in this volume and 7 students in the early stage to discuss their PhD direction.

The paper *Spatial Indexes for Simplicial and Cellular Meshes* (Riccardo Fellegara) addresses the problem of performing spatial and topological queries on simplicial and cellular meshes, by first presenting a family of spatial indexes for tetrahedral meshes and then proposing a specific data structure, for

performing efficient topological queries on simplicial meshes. Extension of the proposed structures to arbitrary dimensions is also discussed.

The paper *Mathematical Methods of Tensor Factorization applied to Recommender Systems* (Giuseppe Ricci, Marco de Gemmis, and Giovanni Semeraro) deals with personalization algorithms able to manage huge amounts of data for the elicitation of user needs and preferences, with a special emphasis on matrix factorization techniques. It also defines a method for tensor factorization suitable for recommender systems.

Finally, the paper *Extended Dynamic Weighted Majority using Diversity to Handle Drifts* (Parneeta Sidhu and MPS Bathia) provides a new framework to handle concept drift for online data, based on the notion of diversity. The resulting online approach guarantees better accuracy with respect to other existing approaches at a slight increase in the running time and memory usage.

8 Conclusions

ADBIS 2013 organizers and ADBIS 2013 satellite events organizers would like to express their thanks to everyone who contributed to the volume content. We thank the authors, who submitted papers to the various events organized in the context of ADBIS 2013. Special thanks go to the Program Committee members as well as to the external reviewers of the ADBIS 2013 main conference and of each satellite event, for their support in evaluating the submitted papers, providing comprehensive, critical, and constructive comments and ensuring the quality of the scientific program and of this volume. We all hope you will find the volume content an useful overview of new trends in the areas of databases and information systems that may further stimulate new ideas for further research and developments by both the scientific and industrial communities. Enjoy the reading!

Part I

ADBIS Short Contributions

New Ontological Alignment System Based on a Non-monotonic Description Logic

Ratiba Guebaili-Djider¹, Aicha Mokhtari¹, Farid Nouioua²,
Narhimene Boustia³, and Karima Akli Astouati¹

¹ RIIMA, USTHB, Algeria

rguebaili@usthb.dz, amokhtari@usthb.dz, kakli@usthb.dz

² Aix-Marseille University, France

farid.nouioua@lsis.org

³ Saad Dahlab University, Algeria

nboustia@gmail.com

Abstract. The choice of the representation formalism of the knowledge manipulated on the Web thanks to the ontologies is a crucial point which can conditioned their use. We must be capable of assuring a maximum of expressiveness in the definition of ontologies' concepts and relations by taking into account the normal context aspect and the exception one. The second very important point is the simultaneous use of several ontologies in a purpose of sharing information. This use became possible thanks to the ontology alignment. It is based on the syntactic, semantic and structural similarities of the different input ontologies. To write the ontology concepts, we propose in this work, a non-monotonic description logic whose semantics is algebraic-based. Then, based on this representation formalism, we show how to improve the measure used to compute the structural similarity.

Keywords: Ontology alignment, Description logic, Normal form, Similarity measure, $OWL_{\delta\epsilon}$, Default and exception, Non-monotonicity.

1 Introduction

In the semantic Web, the structure and the semantics of data are described by means of ontologies, that the software can understand better. Indeed, this mode of representing data facilitates its localization and its integration for various objectives. However, there is no universal ontology, shared and adopted by all users of a given domain. Thus, a key issue to improve system's interoperability is to propose a suitable solution to the heterogeneousness. This can only be done by the reconciliation of the various ontologies used in a domain by the different systems. This reconciliation is often performed by manual or semi-automated ontology integration which consists in identifying the correspondences between concepts from different ontologies. We speak then about ontology mapping (matching, put in correspondences or alignment)[1] [2].

Several ontology alignment approaches exist in the literature (See for example [1],[3], [4],[5]). They are based on various similarity measures.

In a previous paper [6], we proposed an argumentation preference-based system to make decision about the acceptance of concepts correspondances. But, the concepts are written in a liberal form. To improve this representation, so that to ensure a better expression power and to bring more formalization to the process, we have proposed in another work [8], to write concepts of our ontologies on a description logic. But, in commonsense reasoning one often wants to state and to infer relationships that are only "normally" true, but that may have exceptions. This is the main concern of non-monotonic reasoning. For that, in [7] an extension of description logic with the aim of taking into account the non-monotonic knowledge, named $ExtDL_{\delta\epsilon}$ has been proposed. The specificity of our approach is the use of an algebraic-based semantics for our description logic unlike the classical practice where the semantics is based rather on a first order logic interpretation. Following this algebraic semantics, the concepts are written in a particular form called normal form. The principal contribution, in this paper, is the proposition of a new method to compute the structural measure. This measure is not only based on the neighborhood consideration but also on the application of the subsumption operation. This latter is based on the concept normal forms. Our paper will be structured as follows. The section 2, will be dedicated to remind the extension of the description logic named $ExtDL_{\delta\epsilon}$, that we adopted in paper [7], and explain its semantics. In section 3, we define, using our formalism, a new structural similarity measure. Finally, we end by concluding and giving some perspectives for future work.

2 The $ExtDL_{\delta\epsilon}$ Description Logic

2.1 Language

$ExtDL_{\delta\epsilon}$ is a description logic including default's (δ) and exception's (ϵ) connectives for concept definition. $ExtDL_{\delta\epsilon}$ language is inductively defined with a set of primitive roles R, a set of primitive concepts P, constant concept T(Top) and the following syntactic rules.

$C, D \rightarrow \top$ (<i>the most general concept</i>)		\perp (<i>the most specific concept</i>)
P (<i>primitive concept</i>)		$\neg P$ (<i>negation of a primitive concept</i>)
$C \cap D$ (<i>concept conjunction</i>)		$C \cup D$ (<i>concept disjunction</i>)
$\forall R : C$ (<i>value restriction</i>)		$\exists R : C$ (<i>cardinality restriction</i>)
$\geq n$ (<i>maximal cardinality</i>)		$\leq n$ (<i>minimal cardinality</i>)
δC (<i>default concept</i>)		C^ϵ (<i>exception of the concept C</i>)

We use δC to express C as a default concept and C^ϵ as an exception. For example, to express that an elephant is a contemporary animal generally gray and by default has tusks and trunk. A Royal-elephant is a white elephant and is exceptionally not gray. Finally, a dusty royal elephant is a gray Royal-elephant. Formally we write:

Elephant = Animal \cap Contemporary \cap δ Gray \cap δ Tusks \cap δ Trunk
 RoyalElephant = Elephant \cap δ White \cap Gray $^\epsilon$
 DustyRoyalElephant = Elephant \cap White $^\epsilon$ \cap (Gray $^\epsilon$) $^\epsilon$

Let us consider the following equations' set (**EQ**), where A, B and C belong to $ExtDL_{\delta\epsilon}$

$(A \cap B) \cap C = A \cap (B \cap C)$	Prop1.1	$A \cap B = B \cap A$	Prop1.2
$A \cap A = A \cap A = A$	Prop1.3	$\top \cap A = A$	Prop1.4
$A \cap \neg A = \perp$	Prop1.5	$A \cap \perp = \perp$	Prop1.6
$(A \cup B) \cup C = A \cup (B \cup C)$	Prop2.1	$A \cup B = B \cup A$	Prop2.2
$A \cup A = A \cup A = A$	Prop2.3	$\top \cup A = \top$	Prop2.4
$A \cup \neg A = \top$	Prop2.5	$A \cup \perp = A$	Prop2.6
$\delta(A \cap B) = \delta A \cap \delta B$	Prop3.1	$A \cap \delta A = A$	Prop3.2
$A^\epsilon \cap \delta A = A^\epsilon$	Prop3.3	$\delta \delta A = \delta A$	Prop3.4
$(\delta A)^\epsilon = A^\epsilon$	Prop4.1	$(A^\epsilon)^\epsilon = \delta A$	Prop4.2

In the previous example, by replacing the definition of concept "Elephant" in concept "RoyalElephant" and "DustyRoyalElephant" the following definitions hold:

RoyalElephant = Animal \cap Contemporary \cap δ Tusks \cap δ Trunk \cap δ White \cap δ Gray \cap Gray $^\epsilon$ (2.1)

DustyRoyalElephant = Animal \cap Contemporary \cap δ Tusks \cap δ Trunk \cap White $^\epsilon$ \cap δ Gray \cap (Gray $^\epsilon$) $^\epsilon$ (2.2)

- In (2.1) (δ Gray \cap Gray $^\epsilon$) is replaced by (Gray $^\epsilon$) according to Prop3.3
- In (2.2) (Gray $^\epsilon$) $^\epsilon$ is replaced by (δ Gray) according to Prop4.2

The new formulas become:

RoyalElephant= Animal \cap Contemporary \cap δ Trunk \cap δ Tusks \cap δ White \cap Gray $^\epsilon$

DustyRoyalElephant= Animal \cap Contemporary \cap δ Trunk \cap White $^\epsilon$ \cap δ Gray \cap δ Tusks.

2.2 Intentional Semantic

This framework covers different logic aspects of formal concepts' definition and subsumption. In our approach, subsumption is considered from a descriptive and a structural point of view and unlike the classical DL which uses a first order logic based semantics, the associated semantic in our approach is rather algebraic based. For that purpose, We define a structural concept algebra $CL_{\delta\epsilon}$ to give an intentional semantic in which concepts are denoted by the normal form of their properties set as in [7][9][10]. From the class of CL-algebra, we present a structural algebra $CL_{\delta\epsilon}$ which endows $ExtDL_{\delta\epsilon}$ with an intentional semantic. Elements of $CL_{\delta\epsilon}$ are the canonical intentional representation of $ExtDL_{\delta\epsilon}$ terms (i.e. the set of properties presented in their normal forms). We call elements of $CL_{\delta\epsilon}$ normal forms. The definition of $CL_{\delta\epsilon}$ needs a homomorphism h, which

associates an element of $CL_{\delta\epsilon}$ to a term of $ExtDL_{\delta\epsilon}$. Using the equational system given above, we calculate for each concept its single normal form structural denotation. Computing a normal form from a concept description is a “**rewriting**” term based on the equation’s system EQ. Elements of $CL_{\delta\epsilon}$ are 6-tuples pairs with the same structure, the first is used to represent strict properties, the second to default ones.

The sixth field of each 6-tuples represents exception, which includes a 6-tuples’ possibly empty set, where each 6-tuples represents an exception concept (An exception concerns only a default property: See section 2 Prop3.3). Intuitively, the $CL_{\delta\epsilon}$ elements’ structure is defined as follow:

Definition 1: An element of $CL_{\delta\epsilon}$ corresponding to a term T of $ExtDL_{\delta\epsilon}$ is a pair $\prec t\theta, t\delta \succ$, where $t\theta$ and $t\delta$ are the strict and the default parts of T, respectively. $t\theta$ and $t\delta$ are 6-tuples (Dom, Min, Max, π , r, ϵ) defined as follows:

- Dom: is an individuals’ set $\{I_1, \dots, I_n\}$, if the description includes the property ONE-OF $\{I_1, \dots, I_n\}$ else the symbol UNIV (Universe).
- Min (resp. Max): is a real u, if the description includes the property Min u (resp. Max u) else we use the default value MIN-R (resp. MAX-R).
- π is a primitive concepts set of T .
- r is a set of elements defined as follows: $\prec R, c \succ$ where:
 - R is a role name.
 - c is a structure, if T includes $\forall R : c$ in its definition else a denotation of T otherwise.
- ϵ is a set of 6-tuples (Dom, Min, Max, π, r, ϵ).

Notation: the complete structure is denoted by $\prec (t_{\theta dom}, t_{\theta min}, t_{\theta max}, t_{\theta \pi}, t_{\theta r}, t_{\theta \epsilon}), (t_{\delta dom}, t_{\delta min}, t_{\delta max}, t_{\delta \pi}, t_{\delta r}, t_{\delta \epsilon}) \succ$ Examples of the passage from $CL_{\delta\epsilon}$ ’s elements to $ExtDL_{\delta\epsilon}$ ’s elements are giving in Table 1.

Example: $\prec (Univ, Min - R, Max - R, \{Animal\}, \emptyset, \emptyset), (Univ, Min - R, Max - R, \{Animal, Fly\}, \emptyset, \emptyset) \succ$ is a description of $Bird \equiv Animal \cap \delta Fly$.

2.3 Subsumption

There are several equivalent definitions of subsumption relation. The first response to the question: ‘is C subsuming D?’ is by comparing their instances sets. This approach uses an extensional semantics. The second response approach, that we are interested in, compares the properties’ set of concepts C and D. This approach uses an intentional semantic [9]. In our proposal, the two compared concepts are defined by their descriptive normal forms ss stated above. In Algorithm 1, we answer the previous question in two steps: first, we calculate the normal forms of the concepts C and D and that of their conjunction ($C \cap D$), then we compare the result.

Table 1. Subset of Connectors and constants interpretation of $\mathcal{CL}_{\delta\epsilon}$

$ExtDL_{\delta\epsilon}$	$\mathcal{CL}_{\delta\epsilon}$
\top	$\prec (\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \emptyset, \emptyset),$ $(\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \emptyset, \emptyset)\succ$
P	$\prec (\text{UNIV}, \text{MIN-R}, \text{MAX-R}, P, \emptyset, \emptyset),$ $(\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \emptyset, \emptyset)\succ$
$\forall R : C(C \neq \top \text{ et } C \neq \perp)$	$\prec (\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \{\langle R, \emptyset, 0, c_{\theta.dom} , c \rangle\}, \emptyset),$ $(\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \{\langle R, \emptyset, 0, c_{\theta.dom} , c \rangle\}, \emptyset)\succ$
$\forall R : C \text{ et } C \equiv \perp$	$\prec (\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \{\langle R, \emptyset, 0, 0, b_0 \rangle\}, \emptyset),$ $(\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \{\langle R, \emptyset, 0, 0, b_0 \rangle\}, \emptyset)\succ$
$\forall R : C \text{ et } C \equiv \top$	$\prec (\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \emptyset, \emptyset),$ $(\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \emptyset, \emptyset)\succ$
δC	$\prec (\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \emptyset, \emptyset), c_{\delta} \succ$
C^{ϵ}	$\prec (\text{UNIV}, \text{MIN-R}, \text{MAX-R}, \emptyset, \emptyset, c_{\delta}),$ $(c_{\delta.dom}, c_{\delta.max}, c_{\delta.min}, c_{\delta\pi}, c_{\delta\tau}, c_{\delta\epsilon} \cup c_{\delta})\succ$
\perp	b_0

Algorithm 1. Subsumption Algorithm

Require: C and D two concepts description of $ExtDL_{\delta\epsilon}$ **Ensure:** Response “Yes” or “No” to the question “Is C subsumed by D ?” {Compute normal forms}fn(C) \leftarrow Normalization(C)fn($C \sqcap D$) \leftarrow Normalization($C \sqcap D$)**if** fn(C)= b_0 **then** Response \leftarrow “Yes”**else** **if** fn($C \sqcap D$)= b_0 **then** Response \leftarrow “No” **else** {Comparison of the obtained normal forms: We compare each element of C 's normal form with the equivalent element of $(C \sqcap D)$'s normal form} Compare(fn(C) $_{\theta}$, fn($C \sqcap D$) $_{\theta}$, rep1) **if** rep1=“Yes” **then** Compare(fn(C) $_{\delta}$, fn($C \sqcap D$) $_{\delta}$, rep1) Response \leftarrow rep2 **else** Response \leftarrow “No” **end if** **end if****end if**

Our aim in the next section is to present the modifications we propose in order to improve the structural measure. These modifications are possible thanks to the concepts formalism, as well as the subsumption algorithm detailed above.

3 Structural Measure

Generally, to compute the structural similarity value, the structural techniques based on the taxonomy relations (IS-A) are used. It is obtained by using linguistic similarity, as well as the neighboring structure. For every category of an entity to be aligned, one must extract its neighbors (The direct super-entities, direct sub-entities and The sisters' entities). In our previous work [6], the neighbors of the same type (mothers, sons and sisters) are compared. Every obtained value for every type of neighbor is multiplied by an importance factor which specifies its preference with regard to the other neighbors. The sum of the three values gives the structural similarity value. The limit of this proposal is that it requires to recompute syntactic and semantic similarity values for each of the first ontology's concept with each of the second ontology's concept neighbors. This makes complex the associated treatments. To improve the results and reduce the treatment time, we propose the follow three necessary stages in order to calculate the new similarity value based on the normal forms [7].

Step 1. First, the ontologies coded in $OWL_{\delta\epsilon}$ [7] will be presented in a graphical representation (See for example Fig.2). Note that we have four types of relations:

- $(C \rightarrow C')$, represents the hierarchy relation 'IS-A', C is a sub-concept of C'.
- $(C \xrightarrow{r} C')$, indicates that there is a relation 'r' between C and C'.
- $(C \xrightarrow{\delta} P)$, C has by default a property P.
- $(C - \xrightarrow{\epsilon} P)$ C has an exception on (don't possess) a property P.

Each concept of the pair $\langle C1, C2 \rangle$ is rewritten so that to get back all its properties (generally, it is obtained by the conjunction of its ascendants, default and exception). This will allow us to have a complete definition of the concept.

Example. Let us consider two concepts: "Teacher-Temporary replacement" and "Teacher-Phd Student Researcher" belonging to ontologies O1, O2 respectively (Fig.2). The considered couple of concepts is then: \langle Teacher-Phd Student Researcher, Teacher-Temporary replacement \rangle . We have the following definitions: Teacher-Phd Student Researcher = Teacher \cap Researcher \cap Phd Student \cap δ (Not Worker) \cap (Not Worker) $^\epsilon$ = Teacher \cap Researcher \cap Phd Student \cap (Not Worker) $^\epsilon$

Teacher-Temporary replacement = Teacher \cap Temporary replacement \cap Phd Student \cap δ (Not worker) \cap (Not worker) $^\epsilon$ = Teacher \cap Temporary replacement \cap Phd Student \cap (Not worker) $^\epsilon$

Step 2. Computing the normal forms of the concepts. Every concept of the couple will be replaced by its normal form (See section 2).

Example The normal form of \langle Teacher- Phd Student Researcher, Teacher-Temporary replacement \rangle is the following:

$\langle \langle$ (Univ, MIN-R, MAX-R, $\{Teacher, Researcher, PhdStudent\}$, Φ, Φ)(Univ, MIN-R, MAX-R, $\{Teacher, Researcher, PhdStudent\}$, Φ , $\{NotWorker\}$) \rangle , \langle (Univ, MIN-R, MAX-R, $\{Teacher, Temporary - replacement, PhdStudent\}$, Φ , Φ)(Univ, MIN-R, MAX-R, $\{Teacher, Temporary - replacement, PhdStudent\}$, Φ , $\{NotWorker\}$) $\rangle \rangle$.

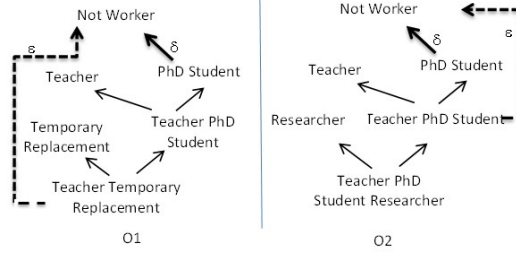


Fig. 1. Two ontologies belonging to a same domain

Step 3. Execution of the subsumption algorithm.

First, we define the concept cardinality as the number of the components of its normal forms. Let us denote by $|C1|$ the cardinality of the normal form of $C1$. $|C1| = \sum |x|$, x is an element of the $C1$ normal form and $|x| \neq 0$ (x must be different from the default value).

Example: $|Teacher- Phd Student Researcher| = 7$. $|Teacher- temporary replacement| = 7$.

After that, we execute the subsumption algorithm for each concept of the couple $(C1, C2)$ as well as the conjunction of concepts $C1 \cap C2$.

Example: We calculate Subsumption (Teacher- Phd Student Researcher, Teacher-Phd Student Researcher \cap Teacher-Temporary replacement) and Subsumption (Teacher-Temporary replacement, Teacher- Phd Student Researcher \cap Teacher- Temporary replacement).

The following cases can occur:

- If the result is $C1$, then we can conclude that $C1 \subseteq C2$. Thus, the similarity value $Sim(C1, C2) = |C1| \div |C2|$
- If the result is $C2$, then we can conclude that $C2 \subseteq C1$. Thus, $Sim(C1, C2) = |C2| \div |C1|$
- If the result is $C1$ and $C2$, the $C1 \equiv C2$. Thus, $Sim(C1, C2) = 1$.
- If the result is equal to ϕ . Then, $Sim(C1, C2) = 0$
- Finally, if the result is different from ϕ , different from $C1$ and different from $C2$, then $Sim(C1, C2) = |C1 \cap C2| \div \max(|C1|, |C2|)$.

Example Subsumption (Teacher-Phd Student Researcher, Teacher-Temporary replacement \cap Teacher-Phd Student Researcher) = Teacher \cap Phd Student \cap (Not Worker)^ε. Thus, the similarity value is: $sim(\text{Teacher-Phd Student Researcher}, \text{Teacher-Temporary replacement}) = |fn(\text{Teacher} \cap \text{Phd Student} \cap (\text{Not Worker})^\epsilon)| \div \max(7, 7) = 5 \div 7 = 0.71$

4 Related Works

In [11], authors propose the writing of the concepts of the biomedical ontologies with default logic, because it admits a transparent representation, and allows a

semantically correct translation to a form of non-monotonic, declarative logic called answer set programs (ASP). In [12], authors suggest endowing OWL of means of description of non-monotonic properties. This extension is based on autoepistemic description logic (ADL) in the context of local closed world reasoning. In [13], a solution of use of several ontologies distributed geographically by integrating within the description logic the default logic is proposed. These approaches are interested in the semantic aspect of an alignment, an essential point which we not treated at the moment, because we are at present interested in the structural aspect of the ontologies. This non-monotonic description logic reasoning hasn't been developed enough (see the recommendation of the Description logic Handbook [14]). Furthermore, the Reiter's default logic [15] had no semantics. It is only long after that, authors in [16] have to show the equivalence with the auto epistemic logic and they thus used its semantics. The semantics of our approach is very simple one and very practical description. It is an algebraic semantics.

5 Conclusion

In this paper, we proposed an ontology alignment system based on the computation of the similarity values. Concepts are not written in a liberal format, i.e., using simply their names, but in a new formalism based on the language of the non monotonic description logic $\text{ExtDL}_{\delta\epsilon}$. Every concept of the ontology is prepresented by its normal form. The specificity of this formalism is the algebraic nature of its semantics. Thanks to this formalism, we proposed a new structural measure to compute the similarity of concepts. In this measure we can give a value to that similarity based on the kinds of relations between entities: equivalence (\equiv), subsumption (\subseteq), incompatibility (\perp), etc. Also, using this concepts rewriting (normal forms), it is possible to take into account other kinds of relations than the relation "IS-A". A default and an exception on concepts' properties can be expressed easily and formally. An implementation of this system is on progress. A first version of the reasoner based on this non-monotonic description logic is already developed. An immediate perspective is the validation of our approach with regard to the reference alignments [9]. A main perspective of our work is to study the modifications of the similarity measures other than the structural one.

References

1. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering* 25(1), 158–176 (2013)
2. Hussain, M.M., Srivatsa, S.K.: A Study of Different Ontology Matching Systems. *International Journal of Computer Applications* (0975-8887) 37(12) (January 2012)
3. Noy, N.F., Musen, M.A.: Algorithm and Tool for Automated Ontology Merging and Alignment. In: *Proc. of the 17th National Conference on A.I (AAAI 2000)*, Austin, Texas, USA, pp. 450–455 (2000)

4. Noy, N., Musen, M.A.: Using Non-Local Context for Semantic Matching. In: Proc. of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001), Seattle, pp. 334–350 (2001)
5. Do, H.-H., Melnik, S., Rahm, E.: Comparison of Schema Matching Evaluations. In: Chaudhri, A.B., Jeckle, M., Rahm, E., Unland, R. (eds.) NODe-WS 2002. LNCS, vol. 2593, pp. 221–237. Springer, Heidelberg (2003)
6. Guebaili_Djider, R., Akli_Astouati, K., Bellili, A., Lacheheb, H.: Cooperative agents for ontology alignment. In: Proceeding of 6th International Conference on Intelligent Interactive Multimedia Systems and Services KES-IIMSS, Sesimba, Portugal, June 26-28 (2013)
7. Guebaili_Djider, R., Mokhtari, A., Boustia, N.: OWL δ ϵ Non Monotonic Ontological Web Language. In: Proc. of 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2012, San Sebastian, Spain, September 10-12 (2012)
8. Guebaili_Djider, R., Mokhtari, A., Boustia, N., Nouioua, F., Akli, K.: A formal language for ontology alignment. In: Proceeding of 5th International Conference on Web and Information Technologie, ICWIT 2013, Tunisia, May 9-12 (2013)
9. Boustia, N., Mokhtari, A.: A dynamic access control model. IN Applied Intelligence Journal 36(1), 190–207 (2012)
10. Coupey, F., Fouquere, C.: Extending conceptual definitions with default knowledge. Comput Intell. 13(2), 258–299 (1997)
11. Hoehndorf, R., Loebe, F., Kelso, J., Herre, H.: Representing default knowledge in biomedical ontologies: application to the integration of anatomy and phenotype ontologies. BMC Bioinformatics (2007)
12. Grimm, S., Motik, B.: Closed World Reasoning in the Semantic Web through Epistemic Operators. In: OWLED 2005, Galway, Ireland (2005)
13. Wiech, P.B.: Distributed Default Reasoning in the Semantic Web. InPhD Thesis, Warsaw University of Technology (2011)
14. Baader, F., McGuinness, D., Nardi, D., Schneider, P.: The description logic handbook: theory, implementation and applications. Cambridge University Press, Cambridge (2008)
15. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13(1-2), 81–132 (1980)
16. Denecker, M., Marek, V.W., Truczynsky, M.: Uniform semantic treatment of default and autoepistemic logics. Artificial Intelligence 143(1), 79–122 (2003)

Spatiotemporal Co-occurrence Rules

Karthik Ganesan Pillai, Rafal A. Angryk, Juan M. Banda,
Tim Wylie, and Michael A. Schuh

Montana State University, Bozeman, Montana-59717
{k.ganesanpillai,angryk,juan.banda,timothy.wylie,
michael.schuh}@cs.montana.edu

Abstract. Spatiotemporal co-occurrence rules (STCORs) discovery is an important problem in many application domains such as weather monitoring and solar physics, which is our application focus. In this paper, we present a general framework to identify STCORs for continuously evolving spatiotemporal events that have extended spatial representations. We also analyse a set of anti-monotone (monotonically non-increasing) and non anti-monotone measures to identify STCORs. We then validate and evaluate our framework on a real-life data set and report results of the comparison of the number candidates needed to discover actual patterns, memory usage, and the number of STCORs discovered using the anti-monotonic and non anti-monotonic measures.

Keywords: spatiotemporal events, extended spatial representations, spatiotemporal co-occurrence rules.

1 Introduction

Spatiotemporal co-occurrence patterns (STCOPs) represent subsets of event types that occur together in both space and time. The discovery of spatiotemporal co-occurrence rules (STCORs) from STCOPs in data sets with evolving extended spatial representations is an important problem for application domains such as weather monitoring, astronomy, and solar physics, which is our application focus, and many others. Given a spatiotemporal (ST) database in which data objects are represented as polygons that continuously change their movement, shape, and size, our goal is to discover STCORs. In this paper we present a novel approach to our recent work initiated in [9], where we introduced the STCOPs mining problem, developed a general framework to discover STCOPs, and introduced an Apriori-based [1] STCOPs mining algorithm. This paper makes the following new contributions: 1) We introduce our novel Apriori-based STCOR-Miner algorithm; 2) We analyse a set of anti-monotonic and non anti-monotonic measures to discover STCORs; and 3) We verify the STCOR-Miner algorithm with a real-life data set, and provide experimental results reporting comparisons on the number of candidate pattern instances and actual pattern instances found for both types of measures, memory usage of the STCOR-Miner algorithm for our measures, and the number of STCORs discovered.

Since ST data mining is an important area, many algorithms have been proposed in literature for co-location mining in ST databases: topological pattern mining [13], co-location episodes [2], mixed drove mining [3], spatial co-location pattern mining from extended spatial representations [14], and interval orientation patterns [8]. However, none of these approaches focus on mining ST co-occurrences from data represented as polygons evolving in time. Due to space constraints we do not give detailed information on these works; however, interested readers read our earlier paper published in [9] for detailed list of literature.

The rest of the paper is organized as follows: We review important concepts of modeling STCOPs for evolving extended spatial representations in Sec. 2. In Sec. 3, we present our proposed STCOR-Miner algorithm. Finally in Sec. 4 we present the experimental evaluation and summary of results.

2 Modeling STCOPs

Given a set of ST event types denoted $E = \{e_1, \dots, e_M\}$, and a set of their instances $I = \{i_1, \dots, i_N\}$, such that $M \ll N$. A STCOP is a subset of event types that co-occur in both space and time.

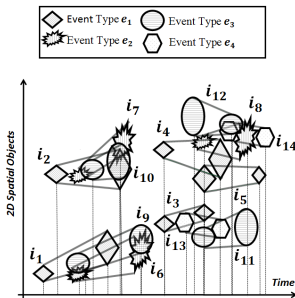


Fig. 1. An ST data set with 2D spatial objects evolving in time

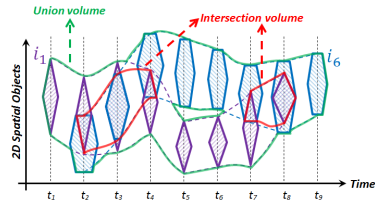


Fig. 2. Example of size-2 co-occurrence of spatiotemporal objects

(a)

Instance ID	Event Type	Start Time (HH-MM)	End Time (HH-MM)
i_1	e_1	10:00	10:30
i_2	e_1	10:10	10:40
i_3	e_1	11:00	11:20
i_4	e_1	11:00	11:30
i_5	e_1	11:20	11:50
i_6	e_2	10:20	10:50
i_7	e_2	10:20	10:40
i_8	e_2	11:20	11:40
i_9	e_3	10:20	10:50
i_{10}	e_3	10:30	10:40
i_{11}	e_3	11:20	11:40
i_{12}	e_3	11:10	11:30
i_{13}	e_4	11:10	11:30
i_{14}	e_4	11:30	12:00

(b)

Instance of e_1	Instance of e_2	Time Instant ($t_s = 10$ minutes)	$Area(i_1 \cap i_6)$	$Area(i_1 \cup i_6)$
i_1	i_6	$t_1 = 10:00$	0	60
i_1	i_6	$t_2 = 10:10$	25	120
i_1	i_6	$t_3 = 10:20$	95	115
i_1	i_6	$t_4 = 10:30$	15	140
i_1	i_6	$t_5 = 10:40$	0	150
i_1	i_6	$t_6 = 10:50$	0	140
i_1	i_6	$t_7 = 11:00$	16	130
i_1	i_6	$t_8 = 11:10$	90	90
i_1	i_6	$t_9 = 11:20$	0	60

$$cce_{i_1 i_6} = \frac{V(i_1 \cap i_6)}{V(i_1 \cup i_6)} = \frac{\sum_{j=t_1}^{t_9} t_s \times Area_j(i_1 \cap i_6)}{\sum_{j=t_1}^{t_9} t_s \times Area_j(i_1 \cup i_6)} = \frac{10 \times (0+25+\dots+90+0)}{10 \times (60+120+\dots+90+60)} = \frac{241}{1005} = 0.23$$

Fig. 3. (a) Table with temporal event information. (b) Example cce calculation.

In Fig. 1, we show an example data set, which we use to explain the concepts in detail. In Fig. 3 a), we show the *Instance ID*, *Event Type*, *Start Time*, and *End Time* of data instances from Fig. 1. This data set contains four event types. The

event type e_1 has five instances, e_2 has three instances, e_3 has four instances, and e_4 has two instances. For simplicity, in this example we do not show the sequence of 2D shapes that reflect the ST evolution of our data.

A *size-k* STCOP is denoted as $SE = \{e_1, \dots, e_k\}$, where $SE \subseteq E$, $SE \neq \emptyset$ and $1 < k \leq M$. We can have multiple *size-k* STCOPs derived from the set E , so to separate them we will use subscripts in future definitions, to denote uniqueness, i.e., $SE_i \neq SE_j$, and SE_i and SE_j contain different event types. *pat_instance* is a pattern instance of an STCOP SE_i , if *pat_instance* contains an instance of *all* event types from SE_i . No proper subset of *pat_instance* is considered to be a pattern instance of SE_i . For example, $\{i_2, i_7, i_{10}\}$ is a *size-3* ($k = 3$) pattern instance of co-occurrence $SE_i = \{e_1, e_2, e_3\}$ in the example data set. A collection of pattern instances of SE_i is a table instance of SE_i , and is denoted as *tab_instance*(SE_i). For example, $\{\{i_1, i_6, i_9\}, \{i_2, i_7, i_{10}\}\}$ is a *size-3* ($k = 3$) *tab_instance*($SE_i = \{e_1, e_2, e_3\}$) in the example data set. An STCOR is of the form $SE_i \Rightarrow SE_j(cce, p, cp)$, where SE_i and SE_j are STCOPs, such that $SE_i \neq SE_j$, and parameters *cce*, *p*, and *cp* characterize the rule in the following manner: (a) *cce* is the ST co-occurrence coefficient and it indicates the strength of ST relation's occurrence that is investigated. The STCOPs mining algorithm [9] uses the ST relation Overlap for *cce*. To distinguish the ST relation from the purely spatial one, we will use, capital letter in the name of the former. Some examples of ST Overlap are $\{i_1, i_6\}$, $\{i_2, i_7\}$, and $\{i_7, i_{10}\}$ in Fig. 1. (b) *p* is the *prevalence measure*. The *prevalence measure* emphasizes how interesting the ST co-occurrences are based on prevalence. In our investigation, we used the participation index (*pi*), proposed in [6], as the *prevalence measure*. The *pi* is monotonically non-increasing when the size of the STCOP increases, which is exploited for computational efficiency [6]. (c) *cp* is the conditional probability [6] of our STCOR. The *cp* gives the confidence of the STCOR $SE_i \Rightarrow SE_j$.

2.1 Measures

Our STCOPs algorithm [9] uses the ST co-occurrence coefficient to calculate *cce*. The ST co-occurrence coefficient is closely related to the coefficient of areal correspondence (CAC) proposed in [12]. CAC is computed for any two or more overlapping polygons as the area of their intersection, divided by the area of union (spatial version of *Jaccard* measure [7]). In [9], we extend CAC to three dimensions (two dimensions correspond to space and the third dimension corresponds to time) and calculate the ST co-occurrence coefficient using ST volumes: (1) The Intersection volume of a *size-k* pattern instance, denoted $V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k)$, is the volume of the three dimensional object representing the Intersection of the trajectories of all instances involved in a given pattern instance. (2) The Union volume of a *size-k* pattern instance, denoted as $V(i_1 \cup i_2, \dots, i_{k-1} \cup i_k)$, is the volume of the three dimensional object representing the Union of the trajectories of all instances involved in a given co-occurrence.

2.2 Co-occurrence Coefficient *cce*

We use the *cce* as our measure to assess the strength of the ST relation Overlap. *cce* is calculated for a *size-k* pattern instance as the ratio $J = \frac{V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k)}{V(i_1 \cup i_2, \dots, i_{k-1} \cup i_k)}$. The symbol J represents the *Jaccard* measure [7] (Fig. 2), which is commonly accepted by data mining practitioners [7,11]. Computing the *cce* for extended ST representations such as evolving polygons is not a trivial task. In Fig. 2, we show the movement of a pair of instances of two event types that change sizes and directions across different time instances. We also show the region of Intersection and the region of Union at different time slots. If we assume that instances $\{i_1, i_6\}$, in our example data set (Fig. 1 and Fig. 3 a)), have ST Intersection volume $V(i_1 \cap i_6) = 241$ and a ST Union volume $V(i_1 \cup i_6) = 1005$, then, $cce = \frac{V(i_1 \cap i_6)}{V(i_1 \cup i_6)} = 0.23$ (see the notes under Fig. 3 b) for detailed calculations).

Coefficients	Formula	Anti-monotonic
<i>Jaccard (J)</i>	$\frac{V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k)}{V(i_1 \cup i_2, \dots, i_{k-1} \cup i_k)}$	Yes
<i>Overlap (OMAX)</i>	$\frac{V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k)}{\max(V(i_1), \dots, V(i_k))}$	Yes
<i>Cosine (N)</i>	$\frac{\sqrt[k]{k \times V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k)}}{\sqrt[k]{\sum_{j=1}^k V(i_j)^k}}$	No
<i>Dice (D)</i>	$\frac{k \times V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k)}{\sum_{j=1}^k V(i_j)}$	No
<i>Cosine (C)</i>	$\frac{V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k)}{\sqrt[k]{V(i_1) \times V(i_2) \times \dots \times V(i_{k-1}) \times V(i_k)}}$	No
<i>Overlap (OMIN)</i>	$\frac{V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k)}{\min(V(i_1), \dots, V(i_k))}$	No

Fig. 4. Measures evaluating ST relation Overlap (*cce*)

Although, we have shown calculation of our *cce* using the *Jaccard* measure, in this paper we would like to investigate alternative measures in detail. We analyze six different measures (denoted in the first column of Fig. 4) to assess the strength of ST Overlap.

2.3 Prevalence of STCOPs

The participation index $pi(SE_i)$ of an STCOP SE_i is $\min_{j=1}^k pr(SE_i, e_j)$, where k is the size of the pattern (cardinality of SE_i), and the participation ratio $pr(SE_i, e_j)$ for an event type e_j is the fraction of the total number of instances of e_j forming ST co-occurring instances in SE_i . For example, from Fig. 1 and Fig. 3 a) we can see that the pattern instances of $SE_i = \{e_1, e_2, e_3\}$ are $\{\{i_1, i_6, i_9\}, \{i_2, i_7, i_{10}\}\}$. Only two (i_1, i_2) out of five instances of the event type e_1 participate in $SE_i = \{e_1, e_2, e_3\}$. So, $pr(\{e_1, e_2, e_3\}, e_1) = 2/5 = 0.40$. Similarly $pr(\{e_1, e_2, e_3\}, e_2) = 2/3 = 0.67$, and $pr(\{e_1, e_2, e_3\}, e_3) = 2/4 = 0.50$. Therefore the participation index of STCOP $SE_i = \{e_1, e_2, e_3\}$ is $pi(\{e_1, e_2, e_3\}) = \min(0.40, 0.67, 0.50) = 0.40$. The STCOP SE_i is a *prevalent pattern* if it satisfies a user-specified minimum participation index threshold pi_{th} . In our example above, if the minimum threshold is set to $pi_{th} = 0.3$, then the STCOP $SE_i = \{e_1, e_2, e_3\}$ is a *prevalent pattern*. The conditional probability $cp(SE_i \Rightarrow SE_j)$ of

an STCOR $SE_i \Rightarrow SE_j$ is the fraction of pattern instances of SE_i that satisfies the ST relation strength indicator cce to some pattern instances of SE_j . It is computed as, $\frac{|\pi_{SE_i}(tab_instance(\{SE_i \cup SE_j\}))|}{|tab_instance(\{SE_i\})|}$, where π is the relational projection operation with duplicate elimination [6].

2.4 Problem Statement

Inputs:

1. A set of ST event types $E = \{e_1, e_2, \dots, e_M\}$ over a common ST framework.
2. A set of N ST event instances $I = \{i_1, i_2, \dots, i_N\}$, each $i_j \in I$ is a tuple $\langle instance-id, event\ type, sequence\ of\ \langle 2D\ shape, time\ instant \rangle\ pairs \rangle$.
3. A user-specified ST thresholds for: cce_{th} , pi_{th} , and cp_{th} .
4. A time interval of data sampling (t_s)(the same for all events).

Output: Find the complete and correct result set of STCORs with $cce > cce_{th}$, $pi > pi_{th}$, and $cp > cp_{th}$.

3 STCOR-Miner Algorithm

In this section we introduce our STCOR-Miner algorithm to mine STCORs from data sets with extended spatial representations that evolve over time. Fig. 5 gives the pseudocode of our STCOR-Miner algorithm. The inputs and outputs are as defined in Sec. 2.5. In the algorithm, steps 1 and 2 initialize the parameters and data structures, steps 3 through 10 give an iterative process to mine STCORs and step 11 returns a union of the results of the STCORs (rules of all sizes). Steps 3 through 10 continue until there is no candidate STCOPs to be mined. Next we explain the functions in the algorithm.

Step 2: In step 2, the evolution of instances of our ST events from their start time slot is projected using t_s (to increment the number of time steps between

Variables :

- (1) k the co-occurrence size (Sec. 2).
- (2) C_k : a set of candidates for size- (k) STCOPs derived from size- $(k-1)$ prevalent STCOPs.
- (3) T_k : set of instances of size- (k) ST co-occurrences (Sec. 2).
- (4) P_k : a set of size- (k) prevalent STCOPs derived from size- (k) candidate STCOPs (Sec. 2).
- (5) R_k : a set of ST co-occurrence rules derived from size- (k) prevalent STCOPs (Sec. 2).
- (6) R_{final} : union of all STCORs (rules of all sizes).

Algorithm :

```

1   $k=1; C_1=E; P_1 = E; R_{final} = \emptyset;$ 
2   $T_1 = gen\_loc(C_1, I, t_s);$ 
3  while ( $P_k \neq \emptyset$ ) {
4     $C_{(k+1)} = gen\_candidate\_coocc(P_k);$ 
5     $T_{(k+1)} = gen\_tab\_ins\_coocc(C_{(k+1)}, cce_{th});$ 
6     $P_{(k+1)} = pre\_prune\_coocc(C_{(k+1)}, pi_{th});$ 
7     $R_{(k+1)} = gen\_rules\_coocc(P_{(k+1)}, cp_{th});$ 
8     $R_{final} = R_{final} \cup R_{(k+1)};$ 
9     $k = k + 1;$ 
10 }
11 return  $R_{final};$ 

```

Fig. 5. STCOR-Miner Algorithm

time slots). The combination of the event instance ID and time step will allow us to identify an event at a particular moment. Step 4 generates candidate STCOPs in this step. We use an Apriori-based [1] approach to generate the candidates of size- $(k + 1)$ using ST co-occurring prevalent patterns of size- (k) for anti-monotonic measures. Hence, prevalent patterns of size- (k) , which satisfy the user-specified threshold value of a minimum participation index pi_{th} , are used to generate candidate patterns of size- $(k + 1)$. However, for correctness, we generate candidate patterns of size- $(k + 1)$ from all size- k patterns for non anti-monotonic measures (Fig. 4). Step 5 generates table instances for candidate patterns of size- $(k + 1)$. Pattern instances for each table instance can be generated by an ST query. The geometric shapes of the instances at each time step are saved, as these geometric shapes will be used for finding the *cce* of STCOPs of size three or more. Pattern instances that have a $cce < cce_{th}$ are deleted from the table instance, if the measure used to calculate *cce* has the anti-monotonic property (i.e., J and $OMAX$). However, for non anti-monotonic measures (Fig. 4), pattern instances that do not have any volume resulting from intersection of trajectories of the event instances, are deleted from the table instances. Step 6 discovers filtered size- $(k + 1)$ STCOPs by pruning $C_{(k+1)}$ that have $pi < pi_{th}$. However, please note, if the measure used to calculate *cce* is non anti-monotonic, we will not prune the candidates based on pi_{th} value. Step 7 generates STCORs. A set of STCORs R that have cp greater than cp_{th} of size- $(k + 1)$ is generated from $P_{(k+1)}$ [6] for anti-monotonic measures. However, for non anti-monotonic measures we generate rules that have cp greater than cp_{th} from patterns of $P_{(k+1)}$ that have pi value greater than pi_{th} (note, this check is necessary for non anti-monotonic measures because we do not prune away patterns, see Step 6). Step 8 calculates the union of rules R_{final} and R_{k+1} . The algorithm runs iteratively until no more STCOPs can be generated for anti-monotonic measures. However, for non anti-monotonic measures all the patterns are generated and in a post processing step only the patterns that satisfy pi_{th} are reported. Finally, the algorithm returns the union of all the found STCORs in Step 11.

4 Experimental Evaluation and Conclusions

In our experiments, we use a real-life data set from the solar physics domain ([5],[10]) which contains evolving instances of six different solar event types observed on 01/01/2012. We investigate STCOR-Miner with the measures to accurately capture the STCORs of solar event types represented as evolving polygons. Moreover, an interesting ordering relation on the selectivity of the boolean versions of J , $OMAX$, N , D , C , and, $OMIN$ measures is shown in [4]. We show the ordering relation of the measures on real numbers in our experiments. For all experiments, the $cce_{th} = 0.01$, $pi_{th} = 0.1$, $cp_{th} = 0.6$, and $t_s = 30$ minutes.

4.1 Conclusion on the Count of Pattern Instances

We first investigated the number (no.) of candidate *pat_instance*'s that are used to generate the pattern instances that satisfy the threshold cce_{th} for

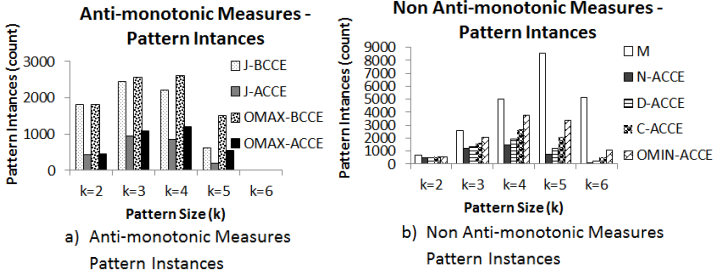


Fig. 6. Pattern Instances

anti-monotonic and non anti-monotonic measures. In Fig. 6 (a), we show the number of pattern instances used by STCOR-Miner with anti-monotonic measures for different pattern sizes. In Fig. 6 (a), J-BCCE (OMAX-BCCE) represent the no. of candidate *pat_instance*'s generated with measure *J* (OMAX), and J-ACCE (OMAX-ACCE) represent the no. of *pat_instance*'s after filtering out the candidates that do not satisfy the threshold cce_{th} in the STCOR-Miner algorithm. From Fig. 6 (a), we can observe that the no. of candidate *pat_instance*'s and actual patterns for the measures *J* and *OMAX* follows the ordering $J \leq OMAX$. In Fig. 6 (b), we show the no. of *pat_instance*'s used by the STCOR-Miner algorithm with non anti-monotonic measures for different pattern sizes. In Fig. 6 (b), M represents the no. of candidate *pat_instance*'s generated (i.e., $V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k) > 0$), and N-ACCE, D-ACCE, C-ACCE, and OMIN-ACCE represent the no. of *pat_instance*'s that satisfy the threshold cce_{th} in the STCOR-Miner algorithm (i.e., the actual patterns that are reported on the output). In comparison to the anti-monotonic measures, we keep the candidate *pat_instance*'s that do not satisfy the threshold cce_{th} for the *N*, *D*, *C*, and *OMIN* measures. Moreover, from Fig. 6 (b) we can observe that the no. of *pat_instance*'s that satisfy the threshold cce_{th} for the measures *N*, *D*, *C*, and *OMIN* follows the order $N \leq D \leq C \leq OMIN$.

4.2 Conclusion on Memory Usage

We now investigate the hard-drive memory usage of the STCOR-Miner algorithm candidate table instances with all the pattern instances generated, and candidate table instances after filtering the pattern instances that do not satisfy cce_{th} . In Fig. 7 (a), we show the memory usage of table instances generated with anti-monotonic measures for different pattern sizes: J-BCCE represents the memory usage of table instances for all pattern instances generated, and J-ACCE represents the memory usage after filtering out the pattern instances that do not satisfy the threshold cce_{th} . As expected, from Fig. 7 (a) we can observe that there is a drop in the memory usage after the pattern instances are filtered by applying the threshold cce_{th} . Furthermore, we can observe that the

memory usage J is more expensive than $OMAX$ due to cost of union geometries needed for the calculation of J . In Fig. 7 (b), we show the memory usage of table instances used by the STCOR-Miner algorithm with non anti-monotonic measures for different pattern sizes. M represents the memory usage of table instances for all the pattern instances generated (i.e., $V(i_1 \cap i_2, \dots, i_{k-1} \cap i_k) > 0$), and N -ACCE, D -CCCE, C -ACCE, and $OMIN$ -ACCE represent the memory usage of table instances with pattern instances that satisfy the threshold cce_{th} in the STCOR-Miner algorithm with the measures N , D , C , and $OMIN$, respectively. However, in comparison to the J and $OMAX$ we do not filter candidate pattern instances for the non anti-monotonic measures. Thus, for N , D , C , and $OMIN$, the no. of candidate pattern instances used to generate patterns of higher sizes is greater than J and $OMAX$.

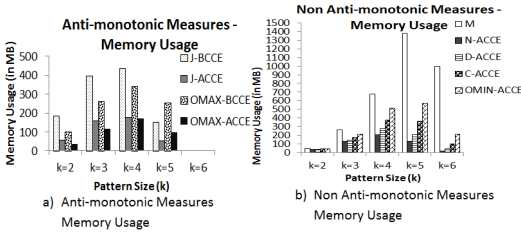


Fig. 7. Memory usage used by candidate table instances

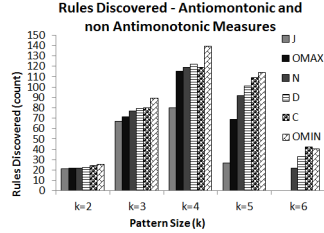


Fig. 8. Number of rules discovered

4.3 Conclusion on Discovered Rules

Finally, we investigate the no. of rules generated using the anti-monotonic and non anti-monotonic measures with the STCOR-Miner algorithm (Fig. 8). The importance of analyzing different measures is shown here in order to accurately capture the ST characteristics of different solar events. For instance, J acts similar to measure D [7]; however, it penalizes objects with smaller Intersection volumes. It gives much lower values than D to objects which have a small Intersection volume - giving a penalty to some of our events that are small in the area and short-lasting. Similarly, the measures $OMAX$ and N also penalize objects with smaller Intersection volume. The measure $OMIN$ [7] gives a value of one if an object is totally contained with another object. We could say that it reflects inclusion, which benefits the objects that are almost equal in space and time. The measure C [7] is more resistant to the size of the objects, making it more appropriate to data sets that contain event types with different life spans and areas (sizes).

Acknowledgements. This work was supported by two National Aeronautics and Space Administration (NASA) grant awards, 1) No. NNX09AB03G and 2) No. NNX11AM13A.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. *SIGMOD Rec.* 22(2), 207–216 (1993)
2. Cao, H., Mamoulis, N., Cheung, D.W.: Discovery of collocation episodes in spatiotemporal data. In: *The 6th Intern. Conf. on Data Mining, DC*, pp. 823–827 (2006)
3. Celik, M., Shekhar, S., Rogers, J.P., Shine, J.A., Yoo, J.S.: Mixed-drove spatiotemporal co-occurrence pattern mining: A summary of results. In: *The 6th Intern. Conf. on Data Mining, DC*, pp. 119–128 (2006)
4. Egghe, L., Michel, C.: Strong similarity measures for ordered sets of documents in information retrieval. *Inf. Process. Manag.* 38(6), 823–848 (2002)
5. HEK (January 2012), <http://www.lmsal.com/isolsearch>
6. Huang, Y., Shekhar, S., Xiong, H.: Discovering collocation patterns from spatial data sets: a general approach. *Trans. on Know. and Data Eng.*, 1472–1485 (2004)
7. Manning, C.D., Schütze, H.: *Foundations of statistical natural language processing*. MIT Press, Cambridge (1999)
8. Patel, D.: Interval-orientation patterns in spatio-temporal databases. In: Bringas, P.G., Hameurlain, A., Quirchmayr, G. (eds.) *DEXA 2010, Part I. LNCS*, vol. 6261, pp. 416–431. Springer, Heidelberg (2010)
9. Pillai, K.G., Angryk, R.A., Banda, J.M., Schuh, M.A., Wylie, T.: Spatio-temporal co-occurrence pattern mining in data sets with evolving regions. In: *ICDM Workshops*, pp. 805–812 (2012)
10. Schuh, M.A., Angryk, R.A., Pillai, K.G., Banda, J.M., Martens, P.C.: A large-scale solar image dataset with labeled event regions. In: *Int. Conf. on Image Processing, ICIP* (2013)
11. Tan, P.-N., Steinbach, M., Kumar, V.: *Introduction to Data Mining*, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston (2005)
12. Taylor, P.: *Quantitative Methods in Geography: An Introduction to Spatial Analysis*. Houghton Mifflin (1977)
13. Wang, J., Hsu, W., Lee, M.L.: A framework for mining topological patterns in spatio-temporal databases. In: *CIKM 2005*, pp. 429–436. ACM, New York (2005)
14. Xiong, H., Shekhar, S., Huang, Y., Kumar, V., Ma, X., Yoo, J.S.: A framework for discovering co-location patterns in data sets with extended spatial objects. In: *SDM* (2004)

R⁺⁺-Tree: An Efficient Spatial Access Method for Highly Redundant Point Data

Martin Šumák and Peter Gurský

P. J. Šafárik University in Košice, Jesenná 5, 04001 Košice, Slovakia
martin.sumak@student.upjs.sk, peter.gursky@upjs.sk

Abstract. We present a new spatial index belonging to R-tree family. Since our new index comes out from the R⁺-tree and holds the concept of non-overlapping nodes we call it R⁺⁺-tree. The original R⁺-tree was designed for both point and spatial data. Using R⁺-tree for indexing spatial data is very inefficient. In our research we face the problem of indexing product catalogues data that can be represented as point data. Therefore we suggested the R⁺⁺-tree for point data only. We present a dynamic index R⁺⁺-tree as an improvement of R⁺-tree. In the tests we show that R⁺⁺-tree offers even better search efficiency than R^{*}-tree when highly redundant point data is considered. Moreover the construction time of R⁺⁺-tree is shorter than the construction time of R^{*}-tree.

Keywords: R⁺-tree, point data, spatial indexing, spatial searching.

1 Introduction

Spatial indexes have been studied for about 30 years. R-trees and its derivatives [1,2,3,4] comprise the most common research branch and they are also used in commercial databases. Our motivation for studying R-trees was driven by the need for efficient computation of top-k query in product catalogues. This article is not about top-k query evaluation it is about new data structure R⁺⁺-tree which is an improved version of R⁺-tree [5]. We suppose it is enough to say that top-k query is similar to k-nearest neighbour query (kNN). While the kNN query uses the distance as a ranking function [7], the top-k query uses more complex function reflecting user preferences [8,9].

It is common in product catalogues, that domains have few possible values of many attributes therefore the redundancy in data is high. Therefore our computational model usually contains top-k query over multidimensional index containing highly redundant point data. Our experiments show that in such scenario R⁺⁺-tree provides the fastest computation time in comparison to all R-tree, R^{*}-tree and R⁺-tree indexes. In our tests we compare R⁺⁺-tree with R-tree, R^{*}-tree and R⁺-tree in top-k query, range query and kNN query search. All search algorithms ideologically work in the same way for all R-trees including R⁺⁺-tree.

2 R⁺⁺-Tree

R-tree, R*-tree and R⁺-tree are designed to store each node on one disk page – each of them with the same fixed size. They all share the same node structure. Leaf entry for an object O is a tuple $(p(O), oid(O))$, where $p(O)$ is the point of object O and $oid(O)$ is an identifier of object O . In other words leaf entry of an object consists of a geometric representation of the object and a pointer to the object possibly residing in external database. Leaf of any of the three trees mentioned above keeps a limited amount of leaf entries. The situation with inner nodes is quite similar. Each inner node keeps a limited amount of inner entries, where each inner entry refers to one child node. Inner entry referring to a child node N is a tuple $(mbr(N), nid(N))$, where $mbr(N)$ is the minimum bounding rectangle of node N (the geometric representation of node N) and $nid(N)$ is an identifier of node N (the pointer to node N).

R-tree, R*-tree and R⁺-tree differ just in the way of construction. Search algorithms over R⁺-tree have to handle possible duplicates, since one object can be stored in several leafs. We offer here just the idea of R⁺⁺-tree design and the dynamic insertion of new object. A thorough description of the dynamic insertion into R⁺⁺-tree is described in full paper version available at http://ics.upjs.sk/~sumak/files/2013_ADBIS_RPP-tree_full.pdf. Our Java implementation of R⁺⁺-tree can be found at <http://ics.upjs.sk/~sumak/files/rpptree.zip>.

2.1 Design of R⁺⁺-Tree

Disadvantage of the original R⁺-tree is the fact that rectangles of child nodes are rarely minimal. Since rectangle of each node has to be completely covered by rectangles of its child nodes, it is impossible to store minimum bounding rectangles only. That is because the use of minimum bounding rectangles causes troubles when adding new object. On the other hand, larger bounding rectangles make the search less efficient. We propose to keep two rectangles for each child node – the minimum one for searching and the larger one for inserting new objects, see Figure 1. This is the basic idea of R⁺⁺-tree – to keep an additional rectangle for each child node, which would actually be the minimum bounding rectangle for related child node. Notation $br(N)$ represents a bounding rectangle of node N but not necessarily the minimum one. Notation $mbr(N)$ represents the minimum bounding rectangle of node. The inner node N of R⁺-tree with parent node P and child nodes M_1, \dots, M_n is: $(nid(P), n, ((br(M_1), nid(M_1)), \dots, (br(M_n), nid(M_n))))$. The inner node N of R⁺⁺-tree is: $(nid(P), n, ((mbr(M_1), nid(M_1)), \dots, (mbr(M_n), nid(M_n))), (br(M_1), \dots, br(M_n)))$. The leaf node with parent node P is the same for both R⁺-tree and R⁺⁺-tree: $(nid(P), n, ((p(O_1), oid(O_1)), \dots, (p(O_n), oid(O_n))))$. Figure 1 shows the representation of inner nodes in the pages of size 4096 B.

The structure of R⁺⁺-tree inner node is designed to keep minimum bounding rectangles together with pointers to child nodes within inner entries. Bounding rectangles are in the second page in the separated list with the same order. Such

representation has the following consequences. Each R⁺⁺-tree inner node takes twice as much space as R⁺-tree inner node, but when searching, the second page does not have to be read. Since leaf nodes have the same structure in both R⁺-tree and R⁺⁺-tree, searching through R⁺⁺-tree requires reading just one page per node (as it is in R⁺-tree). Additional pages do not increase the size of whole tree significantly, because the number of inner nodes is incomparable lower than number of leaves.

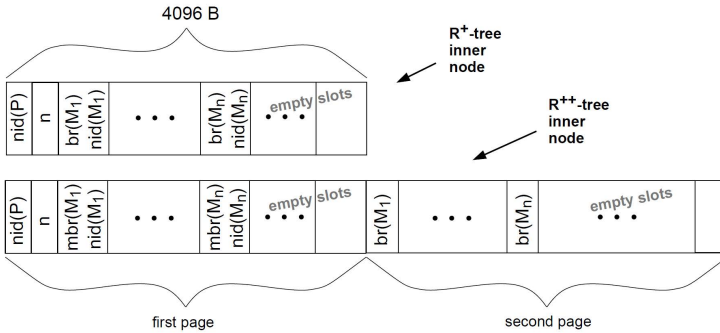


Fig. 1. R⁺⁺-tree inner node always takes two pages, even in the case of low occupancy, when all data would fit in one page

Using this approach, the capacity of R⁺⁺-tree inner node is equal to the capacity of R⁺-tree inner node with the same page size. The additional information stored in second page has to be read only when adding a new object. Beside the structure of inner node, R⁺⁺-tree has its own new algorithm for inserting an object. Basically the splitting method is the only new part. Let us remind that we consider point data only. Before describing the algorithm for object insertion itself, we summarize the facts and properties which hold for R⁺⁺-tree:

1. Leaf node has no occupancy guarantees. Inner node is guaranteed to have at least 1 entry and at least 2 entries if it is the root (node occupancy condition).
2. Bounding rectangle of an inner node completely covers bounding rectangles of its child nodes. Minimum bounding rectangle of an inner node completely covers minimum bounding rectangles of its child nodes. Minimum bounding rectangle of a leaf completely covers points of its objects (nesting condition).
3. Bounding rectangle of an inner node is completely covered by bounding rectangles of its child nodes (complete coverage condition).
4. Bounding rectangle of a node completely covers the minimum bounding rectangle of the node (bounding rectangle vs. minimum bounding rectangle condition).
5. There is no overlap between bounding rectangles of nodes on the same level (zero overlap condition).
6. All leaves are on the same level (balance condition).

2.2 Dynamic Insert

Since all data entries reside in leaves, the first task of inserting a new object is to find an appropriate leaf. Searching a leaf goes down the tree along one path and finds one leaf, in which the new object is going to be added. If object O lies on the boundary of two rectangles, then arbitrary one is chosen. Eventually minimum bounding rectangles along the path are enlarged to encompass the point of a new object. Since complete coverage condition holds true, finding a leaf process never fails in finding an appropriate child node. Since just the point data is considered, minimum bounding rectangles can be enlarged to cover the new point without violation of any condition.

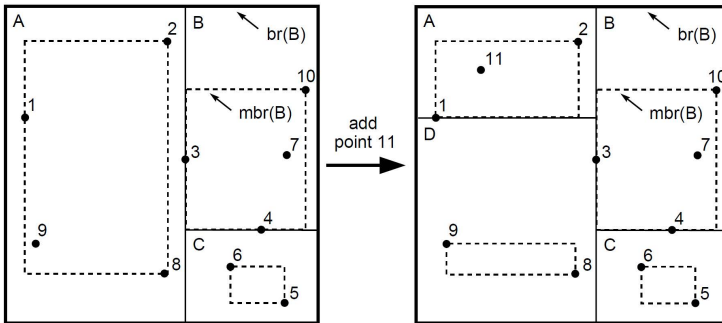


Fig. 2. R^{++} -tree before and after split of leaf node A when adding new object 11. Bounding rectangles are drawn with the full line, the minimum bounding rectangles with the dashed line.

The main issue of inserting process is the splitting of nodes. Basically we use the split algorithm described in [6]. Splitting a leaf (Figure 2) is very simple – the only measure is the balance between number of moved and remained entries. The only problem arises when all objects lie on the same point. In such situation the insert procedure creates an overflow page. In case of many duplicates a chain of overflow pages may occur. Solving such situation by creating a new neighbour leaf and random distribution of entries does not violate the zero overlap condition, because we use point data only. On the other hand it leads to insertion of a new entry into the parent node and possibly to increase of the tree height, which affects search efficiency negatively.

Inserting an object into a leaf may cause an inserting a new entry into its parent inner node. Inserting an entry into an inner node is, if necessary, recursively propagated upwards in the same way. Splitting an inner node is more difficult than splitting a leaf. We evaluate two measures for each tangent hyper-plane to side of a hyper-rectangle of a child node. First of all we try to prevent cuts of child nodes, but it is not always possible to avoid them. Due to the zero overlap condition the cut has to be propagated downward. That may cause a non-optimal cutting of nodes on lower levels. That leads to nodes crumbling,

which affects the searching efficiency negatively. Contrary to the leaf nodes, the cut of inner node is always guaranteed to be found and no overflow pages are necessary. To prove this claim, we have to prove, that each time an inner node is overfilled, there is a hyper-plane with at least one child rectangle lying behind and at least one child rectangle lying in front of this hyper-plane. Since splitting of an inner node always comes after a splitting of a child node, the overfilled inner node contains (amongst others) the original child and the new neighbour child created by the split. Due to this fact the hyper-plane used for splitting the child can be used for splitting its parent, because it is guaranteed that one part of the original child lies behind and the other one in front of the hyper-plane.

2.3 Special Issues of Dynamic Insert

There are two more issues not discussed in previous section. The first one is how to determine boundaries of two nodes arisen from the root split. The second issue is about possibly empty leaf nodes. Let us look at the first issue. After splitting root node the minimum bounding rectangles of its child nodes are easy to compute. The minimum bounding rectangles cannot be used as bounding rectangles residing on the second page of node, unless they together completely cover all domain in each dimension. Since no enlargements of bounding rectangles are allowed in leaf search (only minimum bounding rectangles are enlarged if necessary) the roots children have to completely cover the whole space, where data can appear. We can concrete values when we know the limits of data or we can use infinity values.

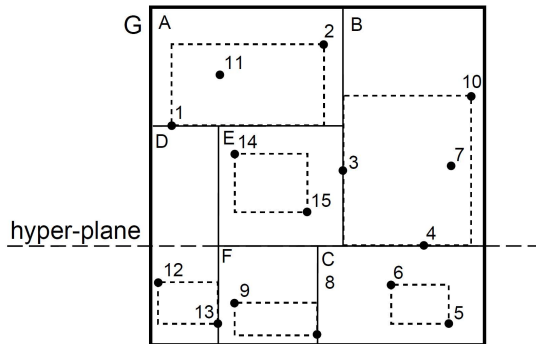


Fig. 3. Leaf node *D* is to be cut by hyper-plane and its part above the hyper-plane is empty

The second issue is empty nodes. Imagine the situation on Figure 3 (splitting of node *G*), where leaf *D* is forced to be cut by a hyper-plane and all of its data fall in front of the hyper-plane. The rest of the leaf *D* behind the hyper-plane is empty and we cannot determine the minimum bounding rectangle of the new

leaf. Since we cannot violate the complete coverage condition, we cannot leave out this node of the tree. We propose to keep this node with the information, that it is empty. Inner node is not designed to keep any information about number of entries residing in its child nodes. Therefore, we propose to use some kind of a null value for the minimum bounding rectangle to determine that related child node is empty.

Search algorithms work exactly the same way they work in R-tree or R*-tree. The only important thing is to read just first pages of inner nodes to use minimum bounding rectangles. However, using bounding rectangles from the second page (i.e. not the minimum ones) does not cause an incorrect search computation, it simply leads to lower search performance comparable to original R⁺-tree.

3 Experiments

Since the proposed R⁺⁺-tree is designed for point data we used several sets of synthetic point data and pseudo-real point data in the tests. We compared R⁺⁺-tree with R-tree and R*-tree. The main measure was the search time efficiency of range query, kNN query and top-k query. We used 4 kB pages for all the tests because it is the size of allocation unit on disks. We used our own Java implementation of R-tree, R*-tree, R⁺-tree and R⁺⁺-tree as well.

In the tests we used the following data distributions: (a, b, c) – synthetic data, (d) – pseudo-real data. Distribution (a) consists of uniformly distributed random points within interval [0; 1] in each dimension and with precision of coordinates to 15 decimal places. Distributions (b), (c) consist of uniformly distributed random points with integer coordinates within interval [0; 100], [0; 10] in each dimension respectively. Distribution (d) is based on real data set containing approximately 27000 flat or house advertisements in Slovakia having 6 attributes: price, area, floor, the highest floor of building, year of approbation and the number of rooms. Values in all 6 attributes are numbers, so we can easily represent each flat by a point in 6-dimensional space. Since the real data set was small, we generated bigger pseudo-real sets by generation of several similar objects for each one from the original set.

For each distribution of synthetic data (a, b, c) we generated 100000 random points with dimensionality from 2 to 10 dimensions, i.e. 27 sets of data altogether. Distribution (a) has almost no redundancy and minimum bounding rectangles of R⁺⁺-tree are almost the same size as the bounding rectangles. The result is that R⁺⁺-tree is very inefficient for all query types. However these data are quite different from real product catalogues data. We do not provide any results in this paper. Distribution (b) has many redundancies in low dimensional spaces and just a few redundancies in higher dimensional spaces. Search performance of R⁺⁺-tree is comparable to R*-tree in all query types for low dimensional spaces. However, even for high dimensional spaces, R⁺⁺-tree is almost always better than R-tree (Figures 4 and 5).

Distribution (c) contains many redundancies in low dimensional spaces. In this case R⁺⁺-tree is the best one. Even in higher dimensional spaces (from 5 to 10

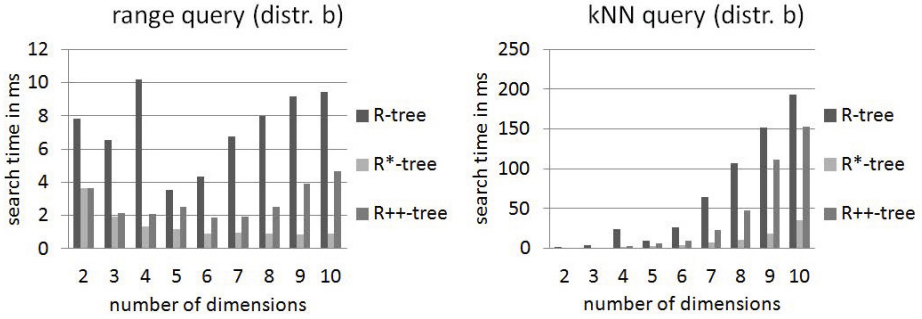


Fig. 4. Average search time (in milliseconds) per range query and kNN query over data with distribution (b)

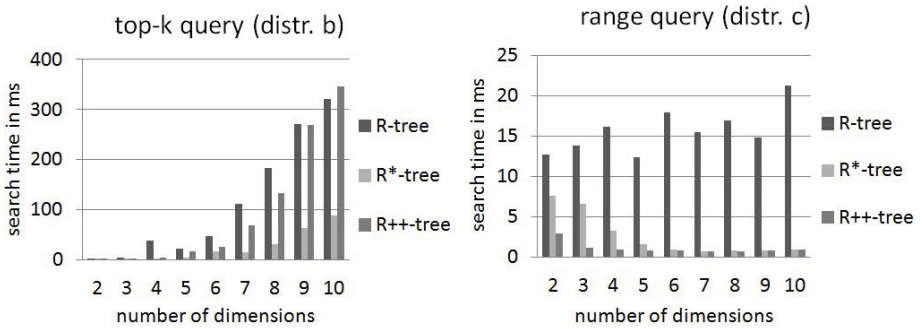


Fig. 5. Average search time (in milliseconds) per top-k query over data with distribution (b) and range query over data with distribution (c)

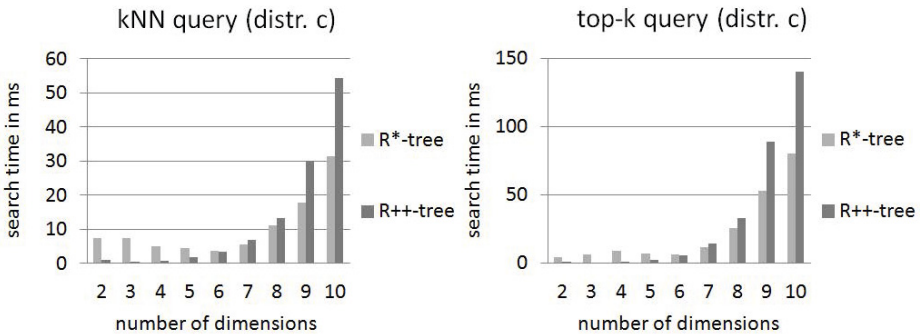


Fig. 6. Average search time (in milliseconds) per kNN query and top-k query over data with distribution (c)

dimensions) R^{++} -tree is comparable to R^* -tree especially in time of range query search. We left out the R -tree of the charts on Figure 6 because its significantly worse results make the differences between R^* -tree and R^{++} -tree illegible.

The results of the tests over pseudo-real data sets are available in full version only. We omitted the R^+ -tree (our implementation according to [6]) from the charts because we found it to be really inefficient.

4 Conclusion

We present the R^{++} -tree as an improvement of R^+ -tree. Even if R^* -tree seems to be universal index and the best in search time in many cases, it falls behind in the efficiency of insertion process. We found out that R^{++} -tree is significantly more efficient than R^* -tree for range query, kNN query and top-k query, when point data with many redundancies is considered. Test results with synthetic data for distribution (c) show that R^{++} -tree is the best up to 4 dimensions. The efficiency of R^{++} -tree slightly decreases with growing dimensionality, because the number of redundancies decreases too. Tests over pseudo-real data also showed that R^{++} -tree offers very good search performance for top-k query, which is the main motivation for our research.

References

1. Guttman, A.: A dynamic index structure for spatial searching. In: SIGMOD Conference (1984)
2. Theodoridis, Y., Sellis, T.: Optimization Issues in R-tree Construction. In: Proceedings of the International Workshop on Geographic Information Systems (1993)
3. Brakatsoulas, S., Pfoser, D., Theodoridis, Y.: Revisiting R-tree Construction Principles. In: Proceedings of ADBIS the 6th East European Conference on Advances in Databases and Information, pp. 149–162 (2002)
4. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R^* -Tree: An efficient and Robust Access Method for Points and Rectangles. In: SIGMOD Conference, pp. 322–331 (1990)
5. Sellis, T., Roussopoulos, N., Faloutsos, C.: The R^+ -Tree: A dynamic index for multi-dimensional objects. In: VLDB (1987)
6. Greene, D.: An Implementation and Performance Analysis of Spatial Data Access Methods. In: Proc. Fifth Intl. Conf. Data Eng., pp. 606–615 (1989)
7. Hjaltason, G.R., Samet, H.: Distance browsing in spatial databases. ACM Transactions on Database Systems, 265–318 (1999)
8. Šumák, M., Gurský, P.: Top-k search in product catalogues. In: Proceedings of DATESO, pp. 1–12 (2011)
9. Šumák, M., Gurský, P.: Top-k search over grid file. In: Proceedings of DATESO, pp. 115–126 (2012)

Labeling Association Rule Clustering through a Genetic Algorithm Approach

Renan de Padua, Veronica Oliveira de Carvalho,
and Adriane Beatriz de Souza Serapião

Instituto de Geociências e Ciências Exatas,
UNESP - Univ Estadual Paulista, Rio Claro, Brazil
paduarenanemail@gmail.com, {veronica,adriane}@rc.unesp.br

Abstract. Among the post-processing association rule approaches, a promising one is clustering. When an association rule set is clustered, the user is provided with an improved presentation of the mined patterns, since he can have a view of the domain to be explored. However, to take advantage of this organization, it is essential that good labels be assigned to the groups, in order to guide the user during the exploration process. Moreover, few works have explored and proposed labeling methods to this context. Therefore, this paper proposes a labeling method, named GLM (*Genetic Labeling Method*), for association rule clustering. The method is a genetic algorithm approach that aims to balance the values of the measures that are used to evaluate labeling methods in this context. In the experiments, GLM presented a good performance and better results than some other methods already explored.

Keywords: Association Rules, Clustering, Labeling Methods, Genetic Algorithm.

1 Introduction

One of the most studied topics in data mining is association mining due to its ability to discover all the frequent relationships that occur among the data set items. The main problem related to this topic is the huge number of patterns that are obtained. Usually, only few of them are of really interesting to the user. To overcome this problem, many approaches have been proposed, being clustering a promising one. In this case, after the rules are obtained, they are grouped in n groups, each one representing a different view of the domain. The idea of the works that use clustering in post-processing is to improve the presentation of the mined patterns, providing the user a view of the domain to be explored, as seen in [1,2,3,4]. However, the grouping becomes useful only if good labels exist, in order to allow an easier browsing of the domain.

Finding good labels is a relevant issue. It is important, for example, that good labels be presented to the user to facilitate exploratory analyses, interesting when the user doesn't have, a priori, an idea where to start. However, few works have explored and proposed labeling methods to the context of association rule

clustering. In [5] a discussion about some methods and their performances, in this context, is done. Since the authors didn't identify any evaluation methodology to check the performance of the methods, two measures were proposed by them. As a result of their study, [5] noticed that none of the methods provide good results in both of the measures, observing that there is a considered difference between their values.

Based on the exposed, this work proposes a labeling method for association rule clustering. The method, named GLM (*Genetic Labeling Method*), is a genetic algorithm approach, since the problem was treated as an optimization one. Its optimization function aims to balance the values of the measures that are used to evaluate labeling methods in this context. In the experiments, GLM presented a good performance and better results than some other methods already explored.

This paper is organized as follows. Section 2 surveys the related works and gives an overview of some labeling methods and evaluation measures. Section 3 describes the proposed method. Sections 4, 5 and 6 present, respectively, the experiments, the results and the conclusions.

2 Background

In this section, works related to the topics covered by this paper are first reviewed. Then, the labeling methods used in this work to compare the performance of GLM and the evaluation measures used in the optimization function are discussed.

Association Clustering. Different clustering strategies have been used for post-processing association rules. In [1] the grouping is done through partitional and hierarchical algorithms using Jaccard, expressed by $J.RT(r, s) = \frac{|\{t \text{ matched by } r\} \cap \{t \text{ matched by } s\}|}{|\{t \text{ matched by } r\} \cup \{t \text{ matched by } s\}|}$, as the similarity measure – the Jaccard between r and s considers the common transactions (t) the rules match. A rule matches a transaction t if all the rule items are contained in t . Furthermore, the authors select as labels of each group the items that appear in the rule which is more similar to all the other rules in the group. In [2] the grouping is done through hierarchical algorithms also using Jaccard as the similarity measure. However, in their work, the Jaccard between two rules r and s , expressed by $J.RI(r, s) = \frac{|\{items \text{ in } r\} \cap \{items \text{ in } s\}|}{|\{items \text{ in } r\} \cup \{items \text{ in } s\}|}$, is computed considering the items the rules share. To label the groups, the same strategy of [1] is used. [4] propose a similarity measure based on transactions and uses a density algorithm to carry out the clustering. In this case, the authors don't mention how the labels are found. [3] also proposes a similarity measure based on transactions, although uses a hierarchical algorithm to carry out the clustering. At the end of the process, the author proposes an approach to summarize each cluster by finding the patterns $a \Rightarrow c$ that cover all the rules in the cluster. Works that combine labeling methods and genetic algorithms, in association context, were not found.

Labeling Methods. The papers related to association rule clustering have not sorely explored the labeling issue, as noticed by [5]. The four labeling methods

presented in [5], used to this context, are briefly described. These methods were used here to allow a comparative analysis of GLM performance. In the *Labeling Method Medoid* (LM-M), the labels of each cluster are built by the items that appear in the rule of the group that represents the medoid of the group. In the *Labeling Method Transaction* (LM-T), the labels of each cluster are built by the items that appear in the rule of the group that covers the largest number of transactions. A rule covers a transaction t if all the rule items are contained in t . In the *Labeling Method Sahar* (LM-S), the labels of each cluster are built by the items that appear in the pattern $a \Rightarrow c$ that covers the largest number of rules. A pattern $a \Rightarrow c$ covers a rule $A \Rightarrow C$ if $a \in A$ and $c \in C$. Finally, in the *Labeling Method Popescul & Ungar* (LM-PU), the labels of each cluster are built by the N items that are more frequent in their own cluster and infrequent in the other clusters.

Evaluation Measures. [5] propose in their work two measures, Precision and Repetition Frequency, that allow an evaluation of labeling methods in the context of association rule clustering. Since any other measures were found to this context, these are the measures used in the fitness function of GLM. Both of the measures range from 0 to 1. Precision (P), expressed by $P(C) = \frac{\sum_{i=1}^{\#Groups} P(C_i)}{\#Groups}$, $P(C_i) = \frac{\#\{rules\ covered\ in\ C_i\ by\ C_i\ labels\}}{\#\{rules\ in\ C_i\}}$, measures how much the labeling method can generate labels that really represent the rules contained in the clusters. It is expected that a good method must have a high precision. However, it is not enough to be precise if the labels appear repeatedly among the clusters. Therefore, Repetition Frequency (RF), expressed by $RF(C) = 1 - \frac{\#\{distinct\ labels\ that\ repeat\ in\ the\ clusters\}}{\#\{distinct\ labels\ in\ the\ clusters\}}$, measures how much the distinct labels that are present in all the clusters don't repeat. The higher the RF value, the better the method, i.e., less repetitions implies in better performance.

3 GLM: The Genetic Labeling Method

GLM is a genetic algorithm approach for labeling association rule clustering. In this proposed labeling method, the labels of the clusters are built by the items that appear in the rules of the groups that ensure a good tradeoff between Precision (P) and Repetition Frequency (RF). Only P and RF were considered since other evaluation measures were not found. Thus, since the problem was treated as an optimization one, the genetic algorithm approach was adopted. The solution was motivated by the fact that none of the methods discussed in [5] provided good results, at the same time, in P and RF . Thereby, a method that yields ways to maximize interesting evaluation measures is a promising one. To understand GLM, the description of the genetic operators and other important aspects are following discussed. For details about the concepts see [6].

Encoding. In GLM, each individual represents a possible solution to the problem, i.e., the labels of each group in an association rule clustering. For that, each individual is composed by n chromosomes, where n represents the number of groups in the clustering given as input. Each chromosome has m genes, where m represents the maximum number of labels to be assigned to each group. m

is a value informed by the user. Although all the chromosomes have the same length m , m is the maximum number of labels a group can have. Thus, in some chromosomes, not all of its genes are filled.

Initialization. Given an association rule clustering, an initial population is generated. A population is composed by PS individuals, where PS (*Population Size*) is given by the user. To create each individual a looping is done, where each iteration is related to a chromosome (group). The choice of the items to be selected as labels (genes), in each chromosome, is done randomly. However, only the items that appear in the rules of the current group are considered. During this process, it is assured that a group (chromosome) can not contain repeated items in its labels (genes).

Genetic Operators. The genetic operators used in GLM, as well, the fitness function and the termination criterion are described below.

A. Selection. The roulette wheel is used to select two individuals to obtain an offspring. For that, the fitness of each individual is considered as its chance to be selected. The higher the fitness the higher the probability an individual has to be selected.

B. Crossover. The uniform crossover is used to obtain an offspring. The unique offspring is generated from the parents with the help of a bit mask, which is obtained for each chromosome. The bit mask is a sequence of 0's and 1's, which indicates from which parent the gene has to be copied. When the value is 0, the offspring inherits the gene from parent 1 and when is 1 from parent 2. Thus, the resulting offspring contains a mixture of genes from both parents. The bit mask is randomly obtained as a vector of bits: when one parent has more filled genes (labels) than the other, the bit mask in these not overlapped positions receives the code related to the filled parent. This fact justifies our choice to obtain a unique offspring: if two offspring were generated, they would be very similar to their parents.

C. Mutation. In offspring, the genes of chromosomes occasionally change with a probability MP (*Mutation Probability*). Only one gene of each chromosome has a chance to be mutated. Thus, for each chromosome, a probability is randomly obtained and compared with MP to check if the mutation will occur in the chromosome. If so, a gene in the chromosome is randomly chosen and the mutation is done.

D. Fitness Function. Since GLM aims to obtain labels that ensure a good tradeoff between Precision (P) and Repetition Frequency (RF), the fitness function of an individual I is defined by $Fitness(I) = (P+RF) - \left(\frac{Max(P,RF)}{Min(P,RF)} * 10^{-5} \right)$. Initially, P and RF are added. However, as 1.0 can be obtained by $P = 0.2$ and $RF = 0.8$ or by $P = 0.5$ and $RF = 0.5$, for example, it is necessary to penalize individuals that present a high variation between the measures to ensure a good tradeoff. The normalized penalization adopted in this work is obtained dividing the measure that has the maximum value (Max) by the one that has the minimum (Min) and, then, normalizing the result with 5 digits of precision (10^{-5}). 10^{-5} represents the ratio $\frac{0.00001}{1.00000}$, in which 0.00001 indicates the minimum value a measure can reach and 1.00000 the maximum. As mentioned before, only P

and RF were considered to define the fitness function, since other evaluation measures were not found to this context. However, as new measures arise, they can also be added to GLM.

E. Termination. GLM stops when the number of iterations, i , is larger than a given number NG (Number of Generations).

The GLM steps are presented in Algorithm 1. The algorithm receives 5 parameters: (i) an association rule clustering (ARC); (ii) the population size (PS); (iii) the mutation probability (MP); (iv) the number of generations (NG); (v) the maximum number of labels to be assigned to a group (m). m gives, in fact, the number of genes the chromosomes will have. At the end of the process, the clustering given as input is outputted to the user with its labels. As seen in Algorithm 1, GLM works as follows: initially, the ARC is loaded to the memory (line 1). After that, a population is created with PS individuals (line 2). Until the stopped criterion is not reached (line 3), the operators of selection (line 4), crossover (line 5) and mutation (line 8) are applied. Before starting a new iteration, the population is updated (line 11), i.e., the offspring is added and the parent with the lowest fitness removed. In the end, the individual with the best fitness represents the solution.

Algorithm 1. The GLM steps.

Input: ARC , PS , MP , NG , m .

Output: A labeled association rule clustering.

```

1: Read  $ARC$ 
2: Initialize population with  $PS$  individuals
3: for 1 to  $NG$  do
4:   Select two individuals  $I_1$  and  $I_2$ 
5:   Crossover  $I_1$  and  $I_2$  to obtain an offspring  $O$ 
6:   for each  $O$  chromosome do
7:     if ( $RN < MP$ ) then
8:       Mutate  $O$  chromosome, where  $RN$  is a random number in the range [0,1]
9:     end-if
10:  end-for
11:  Update population: Add  $O$  to population; Remove the parent ( $I_1;I_2$ ) with the lowest fitness
12: end-for

```

4 Experiments

Some experiments were carried out in order to analyze GLM performance (GLM's quality assessment). Thus, initially, it was necessary to generate some association rule clusterings (ARC). Forty organizations were selected to obtain 40 ARC s for each one of the four data sets used.

The four data sets were Adult (48842;115), Income (6876;50), Groceries (9835;169) and Sup (1716;1939). The numbers in parenthesis indicate, respectively, the number of transactions and the number of distinct items in each data set. The first three are available through the package "arules"¹. The last one was donated by a supermarket located in São Carlos city, Brazil. All the transactions in Adult and Income contain the same number of items (named here as

¹ <http://cran.r-project.org/web/packages/arules/index.html>.

standardized data sets (SDS)), different from Groceries and Sup (named here as *non-standardized data sets* (NSDS)), whereupon each transaction contains a distinct number of items. Thus, the experiments considered different data types. The rules, in each data set, were mined using an *Apriori* implementation² with a minimum of 2 and a maximum of 5 items per rule. With the Adult set 6508 rules were extracted using a minimum support (min-sup) of 10% and a minimum confidence (min-conf) of 50%; Income 3714 rules with min-sup=17%, min-conf=50%; Groceries 2050 rules with min-sup=0.5%, min-conf=0.5%; Sup 7588 rules with min-sup=0.7%, min-conf=0.5%.

To cluster these four rule sets, forty organizations were selected, which one obtained by the combination of an algorithm, a similarity measure and a value of k (number of groups to be obtained). For that, two algorithm (PAM; Ward), two similarity measures (J.RI; J.RT (Section 2)) and ten values of k were considered (5 to 50, steps of 5) ($40=2*2*10$). An organization provides a different way to organize the extracted patterns. In fact, these forty organizations can be grouped in four sets, each one related to a combination of an algorithm and a similarity measure ($2*2$). Although it is necessary to set k , to obtain an organization, this value can be used to analyze the combinations of algorithms and similarity measures on different views. This was the idea used to do the analysis of the results (Section 5). Most of the experiments choices were done based on [5] work.

Before definitely executing GLM and the methods described in Section 2 (LM-M, LM-T, LM-S, LM-PU), in order to do a comparative analysis of its performance, it was necessary to find out the most suitable parameters to set GLM. For that, many experiments were executed to adjust the parameters PS , NG and MP . In each experiment, GLM was executed on all the 40 *ARC*s, in each data set. Being a genetic approach, GLM doesn't obtain the same results for the same parameters every run. Thus, each one of the experiments was executed 10 times in order to obtain an average performance. In the end, the following values were selected: $PS = 50.000$, $NG = 50.000$ and $MP = 0.75$. Regarding the value of m , the parameter was set to 5 (N in LM-PU was set to 5 too). To allow the comparative analysis, the methods LM-M, LM-T, LM-S and LM-PU were also executed on all the 40 *ARC*s, in each data set.

5 Results and Discussion

Since the GLM optimization function aims a good tradeoff between P and RF , all the results are shown and discussed over these measures (only the *ARC*s results related to the GLM selected parameters were considered). Table 1 presents the averages of P and RF in GLM and in the methods used for comparison. Each average was obtained from the results related to the presented configuration. The value $P = 0.740$ in GLM at SDS:PAM:J.RI, for example, was obtained from the average of the P values in GLM at Adult:PAM:J.RI and Income:PAM:J.RI over the k s. Thus, notice that the forty organizations, related to each data set, were grouped in four sets, considering that k can be used to analyze the combinations

² <http://www.borgelt.net/apriori.html> [Christian Borgelt's Web Page].

Table 1. Performance of the labeling methods, measured through P and RF , in the different data types

Data type	Alg.	Sim. M.	LM-M		LM-T		LM-S		LM-PU		GLM	
			P	RF	P	RF	P	RF	P	RF	P	RF
SDS [Adult/ Income]	PAM	J.RI	0.999▲	0.153✓	0.961	0.260	0.965	0.272	0.999▲	0.170✓	0.740✓	0.503▲
		J.RT	0.995	0.355	0.934	0.455	0.965	0.427	0.998▲	0.403✓	0.878✓	0.613▲
	Ward	J.RI	0.996▲	0.338✓	0.915	0.437	0.963	0.423	0.993	0.369	0.847✓	0.557▲
		J.RT	0.988	0.350	0.929	0.535	0.963	0.401	0.995▲	0.412✓	0.912✓	0.616▲
NSDS [Groce- ries/Sup]	PAM	J.RI	0.979	0.511	0.852	0.482	0.913	0.398	0.986▲	0.523✓	0.478✓	0.744▲
		J.RT	0.911	0.611	0.743	0.633	0.818	0.646	0.935▲	0.671✓	0.452✓	0.769▲
	Ward	J.RI	0.955	0.770	0.905✓	0.855▲	0.931	0.787	0.966▲	0.572✓	0.616	0.687
		J.RT	0.899	0.616	0.773	0.690	0.832	0.672	0.929▲	0.645✓	0.698✓	0.704▲

of algorithms and similarity measures on different views (Section 4). Therefore, each presented configuration represents the average of twenty results (10 related to each data set of the same data type).

A comparative analysis was done to evaluate the performance of GLM, in relation to the other methods usually used, based on the average of each measure (P ; RF), considering the different data types, apart from the data set used. For that, in Table 1, the highest averages, regarding each one of the measures (P ; RF), are marked with ▲ in each considered configuration. For the SDS:PAM:J.RI configuration, for example, the best average for RF is the one related to GLM (0.503). In the table, for each ▲, there exist a ✓ on the other measure of the pair P/RF to indicate the method is, in theory, suitable. The measure marked with ▲, in the ▲/✓ pair, indicates the one that leads to the selection of the method – RF in GLM (0.503), for example. Thereby, it is possible to observe, in each considered configuration, the method that presents the best performance. Finally, since the results related to LM-M, LM-T, LM-S and LM-PU are deterministic and the differences among the 10 GLM executions were too small, no statistical test was done. It can be noticed that:

Configurations related to SDS. In all the SDS configurations, the method that presents the best result in RF is GLM and in P LM-M (SDS:PAM:J.RI; SDS:Ward:J.RI) and/or LM-PU (SDS:PAM:J.RI; SDS:PAM:J.RT; SDS:Ward:J.RT). However, it can be observed, in the selected methods (▲/✓ pairs), that P presents good results, different from RF in LM-M and/or LM-PU, where lower values are obtained. Therefore, GLM is a suitable method to be used when seeking for a balance between P and RF , since it improves RF while maintains P .

Configurations related to NSDS. In most of the NSDS configurations, the method that presents the best result in P is LM-PU and in RF GLM (NSDS:PAM:J.RI; NSDS:PAM:J.RT; NSDS:Ward:J.RT) (exception to NSDS:Ward:J.RI with the selection of LM-T for RF). However, it can be observed, in the selected methods (▲/✓ pairs), that P presents better results in LM-PU with a reasonable RF compared to GLM P and RF . Therefore, LM-PU is a suitable method to be used when seeking for a balance between P and RF , since it presents a good P while maintains a reasonable RF . In relation to NSDS:Ward:J.RI, while P presents good results both in LM-PU and LM-T, the

same doesn't occur in LM-PU RF , where a lower value is obtained. Therefore, in this last case, LM-T is the one to be chosen.

Based on the obtained results, it can be noticed that while the method that seems to be more suitable for SDS regarding association rule clustering is the proposed one, i.e., GLM, for NSDS, in almost all the cases, is LM-PU, although GLM presented reasonable results. Thus, GLM seems to be useful in some circumstances and good and useful if a tradeoff between P and RF is essential (it can be seen, from Table 1, that GLM presents good results in almost all the considered configurations).

6 Conclusions

This paper proposed a labeling method for association rule clustering, named GLM, based on a genetic algorithm approach. This is an essential issue, since good labels must be assigned to the groups in order to guide the user during the exploration process. GLM was modeled to balance the values of P and RF , two measures that are used to evaluate labeling methods in this context. In the experiments, GLM presented a good performance and better results than some other methods already explored in the literature, mainly when applied in SDS. As future works, other genetic operators can be tested, as other ways to iterate the population during the process, aiming to refine GLM performance.

Acknowledgments. We wish to thank Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (process number 2010/07879-0) for the financial support.

References

1. Reynolds, A.P., Richards, G., de la Iglesia, B., Rayward-Smith, V.J.: Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms* 5(4), 475–504 (2006)
2. Jorge, A.: Hierarchical clustering for thematic browsing and summarization of large sets of association rules. In: 4th SIAM International Conference on Data Mining, pp. 178–187 (2004)
3. Sahar, S.: Exploring interestingness through clustering: A framework. In: IEEE International Conference on Data Mining, pp. 677–680 (2002)
4. Toivonen, H., Klemettinen, M., Ronkainen, P., Hättönen, K., Mannila, H.: Pruning and grouping discovered association rules. In: Workshop Notes of the ECML Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases, pp. 47–52 (1995)
5. Carvalho, V.O., Biondi, D.S., Santos, F.F., Rezende, S.O.: Labeling methods for association rule clustering. In: 14th International Conference on Enterprise Information Systems, pp. 105–111 (2012)
6. Sivanandam, S.N., Deepa, S.N.: *Introduction to genetic algorithms* (2008)

Time Series Queries Processing with GPU Support*

Piotr Przymus¹ and Krzysztof Kaczmariski²

¹ Nicolaus Copernicus University, Poland
eror@umk.mat.pl

² Warsaw University of Technology, Poland
k.kaczmariski@mini.pw.edu.pl

Abstract. In recent years, an increased interest in processing and exploration of time-series has been observed. Due to the growing volumes of data, extensive studies have been conducted in order to find new and effective methods for storing and processing data. Research has been carried out in different directions, including hardware based solutions or NoSQL databases. We present a prototype query engine based on GPGPU and NoSQL database plus a new model of data storage using lightweight compression. Our solution improves the time series database performance in all aspects and after some modifications can be also extended to general-purpose databases in the future.

Keywords: time series database, lightweight compression, data-intensive computations, GPU, CUDA.

1 Introduction

Time Series analysis plays a crucial role in many important computational applications. Various hardware or software which must be monitored in order to ensure proper level of service quality emit time series as self describing data. This kind of *machine generated databases* are often growing with square factor since they represent monitoring relations between a distributed system's components in 'all-to-all' fashion. Number of time series generated in this way grows very quickly and falls under category of *Big Data*: problems in which size of data is a problem itself. Classical statistical systems (like R or SAS [7,2]), although capable of performing advanced analysis, are no longer able to handle the newly appearing challenges:

- Very large volumes of data must be consumed by a database in real time. If 1000 machine reports 1000 values every 10 seconds the systems must store $8.64 \cdot 10^9$ data points every day. Because of continuous operation of the system there is no possibility of batch processing.
- Resolution of data cannot be lost. Industrial systems benefit from ability to track single events as well as global tendencies or changes. Correlations between quickly appearing events cannot be found on general level.
- System must be able to answer any kind of queries to the database in reasonable time even if a query involves billions of points. This tight efficiency constrain may be only fulfilled if computation power is not bounded and scales well, possibly linearly.

* The project was funded by National Science Centre, decision DEC-2012/07/D/ST6/02483.

- Storage may not be limited and should scale transparently. SQL databases with centralized indexes are no longer sufficient for these requirements or cannot meet limited budget requirements.

New systems like OpenTSDB [4] or Tempo-DB [6] try to address the above needs by using *Big Table* [8] data model. They are able to import data very efficiently while distributing it in a cloud-like storage. Querying is done by retrieving fragments of data from the distributed regions and putting them together with a map-reduce algorithm. One of the bottlenecks for a time series database is IO bandwidth and centralized aggregation process. Query processing for a longer period of time may need to process hundreds millions of data points. In such cases system reaction time often becomes too long.

1.1 General-Purpose Computation on Graphics Processing Units (GPGPU)

GPU programming offers tremendous processing power and excellent scalability with increasing number of parallel threads. However, vector-like processing in GPU has some limitations. One of them is obligatory data transfer between RAM (random-access memory) of the host machine and the computing GPU device, which generates additional cost when compared to a pure CPU-based solution. This barrier can make GPU-based algorithms unsatisfactory especially for smaller problems. One of the goals of this work is to improve efficiency of data transfer between disk, through RAM, global GPU memory and processing unit.

One of the possibilities, and often the only option, to optimize the mentioned data transfer is to reduce its size by compressing. Classical compression algorithms are computationally expensive (gain from the transfer data does not compensate the calculations [18]) and difficult to implement on the GPU [16]. Alternatives such as lightweight compression algorithms which are successfully used for CPU and GPU are therefore very attractive [18,10,12].

This paper addresses optimizations in time series systems like OpenTSDB allowing for faster query response. We present a new model of data storage using lightweight compression and parallel query processing using GPU. The rest of the paper is organized as follows. In the rest of this section we motivate and presents some of the related works concerning time series, big data and GPU processing. In section 2 we explain the prototype system architecture and querying process, while in section 3 a reader may find experimental results. Section 4 concludes.

1.2 Motivation

There are evidences of configurations pushing tens of billions data points a day into a monitoring system (like Facebook or Twitter). In such complicated cases system often stores very detailed measurements taken in different metrics and configurations for example every 10 seconds and therefore must deal not only with many points in the same time series but also with a huge number of time series as well.

What we observed in our industrial experience is that a user often performs many different queries working on the same time series in the fixed period of time. The reason for this is that users want to observe the same point in time from many angles, which

means performing different types of aggregations of different dimensions. Obviously, this analysis strategy cannot be predicted and aggregations cannot be preprocessed. OpenTSDB saves data points in cache in order not to repeat very expensive hbase scan operation. However, serialized points aggregation was noticed to be slower for large number of time series. Therefore, we propose to use GPU as an alternative query coprocessor using our novel lightweight time series compression strategy. What is more important many users already own powerful graphical devices which may be used as local coprocessors for data analysis. Database querying should take into account this new possibility of supporting time consuming computations.

The main motivation of this work is to open new possibilities in time series querying by utilization of GPU processors for ultra fast time series aggregation on both server and client side. In this paper we also show that GPU processor may be used to perform computations on compressed data without introducing any additional costs. This in turn allows for application of GPU processors not only in computation-intensive problems in which time of copying data is amortized by numerical computations but also in data-intensive problems. This achievement opens a new filed for general database algorithms. Our solution improves the overall time series database performance by: minimizing communication footprint between a data storage and a query engine (by using i.a.: data compaction and lightweight compression) and moving data decompression and time series query processing to GPU.

1.3 Related Works

There is huge interest in efficient time series processing both in industry and science since large (and growing fast) data sets need to be queried and stored efficiently. OpenTSDB [4] build on top of HBase [1] and offers tremendous speed of data insertion and scanning together with high scalability. However, its data model is limited and so far cannot handle many important cases (like data annotations) Unde et al. [15] claim that OpenTSDB reaches much better performance than DB2 RDBMS for time series processing. Our experiments showed that OpenTSDB performance degrades if there are more than just a few tags attached to single metric which means that it has to aggregate too many time series.

Real time data analytic is offered by ParStream [5] data base system using GPU nodes. Data is equally distributed along machines. Combination of CPU and GPU processing together with load balancing enables to achieve almost real time processing of terabytes of data [11]. Also Jedox [3], an OLAP database system, offers possibility of using GPU as a coprocessor in queries. Both solutions are not strictly focused on time series processing and therefore probably cannot offer many optimizations which could be potentially possible. Our research is aimed at similar goals but with stress on large number of time series.

In [17] authors present interesting study of different compression techniques for WWW data in order to achieve querying speed-up. A general solution for data intensive applications by cache compression is discussed in [18]. Obviously the same technique may be used for time series and the efficiency may be increased if decompression speed is higher than I/O operation. In this paper we also show that decoding is really much faster and eliminates this memory bottleneck. The compression schemes proposed by

Zukowski et al. [18] offer good trade-off between compression time and encoded data size and what is more important are designed especially for super scalar processors which means also very good properties for GPU.

2 System Architecture

This section presents our system architecture. A set of collectors performs some treatment of source data and send it to the storage. Then data is sent to a storage which can be easily realized by column based NoSql database systems like Hbase [1] or Cassandra [9]. During the insertion process time series are compacted into larger records (taking into account the metric name and tags) containing a specified period of time (eg 15 minutes, 1 hour, 2 hours, 24 hours – depending on the number of observations) which differs from OpenTSDB design. The last important part of the system is the query engine responsible for user-database interactions. Again it must retrieve and process data in time acceptable for a user even if queried time period is long and covers many time series and data points. Following other solutions, as an answer to this problem we propose a analytic coprocessor but with GPU computing support. Since data transfer time is critical for distributed systems the key improvement over any other time series solution is decreasing size of necessary data transfer. In our solution we combine storing data internally compressed with adapted lightweight compression and ultra fast decompression done directly into GPUs memory. This strategy minimises not only storage size but also significantly increases transfer speed and processing time. In the last stage, utilization of GPU allows for very fast query processing.

2.1 Query Processing

The overall process of query execution is shown in Fig. 1a, while detailed query processing steps are presented below.

Decompression: OpenTSDB uses HBase support for lightweight compression algorithms such as Snappy or LZO. However, our observations suggest that the use of specialized lightweight compression algorithms like PFOR and PFOR-DIFF can significantly raise performance. Moreover, lazy decompression can be considered as a one of the stages of query processing, which minimizes the cost of memory transfers. Results are further improved by ultra fast decompression done by GPU processor. Obviously, better compression coefficients can be obtained due to the well-known characterization of the stored data. In this work we use modified PFOR and PFOR-DIFF from our earlier work [12].

Quantization: An important aspect is the analysis of the data at different levels of detail. This means that we have the opportunity to analyse the long-term general aspects as well as short-term detailed ones. Moreover, it allows us to limit the number of details in data, and thus reduce the initial size of it prior to processing. This important part of query processing may be efficiently performed on GPU: each thread examines j data elements (Fig. 1b). In a loop, it makes the quantization of the time series. Quantization is carried out using threads buffers to reduce the number of atomic operations needed for global memory. In the end, the partial results are stored in memory using global atomic operations.

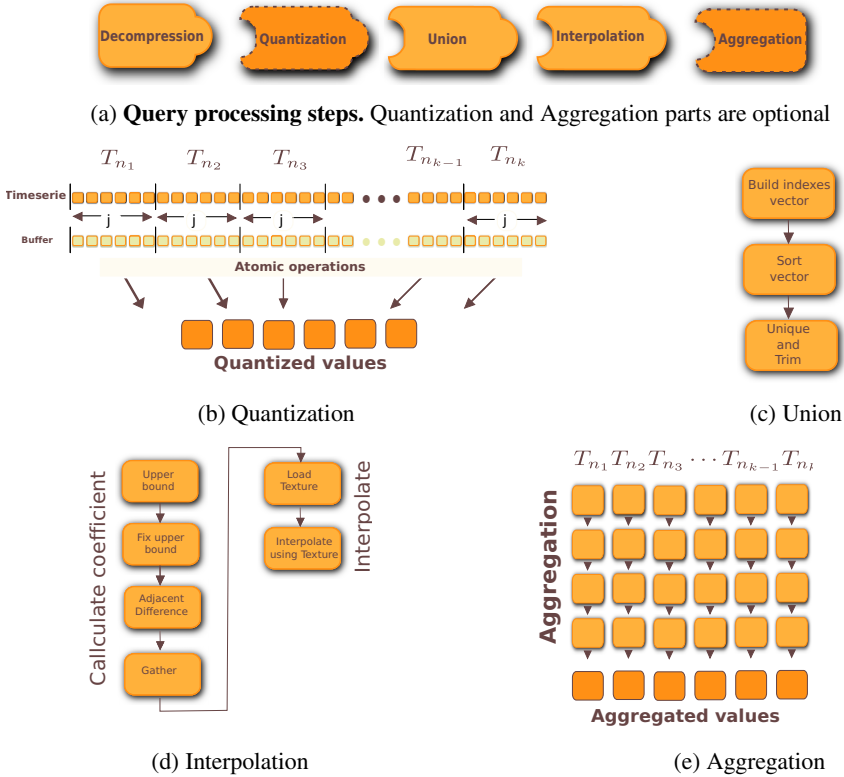


Fig. 1. Query operations (where $T_{n_1}, T_{n_2}, \dots, T_{n_{k-1}}, T_{n_k}$ are threads)

Union: In order to calculate aggregation for non evenly sampled time series, we need to transform them into evenly sampled ones (through interpolation). The first step is to determine a set union of timestamp indices for all time series. Again this stage can be efficiently implemented using *Thrust* GPU library offering basic operations for vectors (the interface is similar to the Standard Template Library vector for C++). In the first step, we build the vector of time series. Then the timestamps are sorted using sort method – which performs highly optimized Radix Sort. Subsequently unique operation is performed which removes duplicate items. See outline in Fig. 1c.

Interpolation: In the previous step we calculated the union of timestamp indices (t_i). Here, we need to interpolate values for selected timestamps in every time series. Finally, we obtain an evenly sampled time series. To improve efficiency of this part we used textures with CUDA hardware support for linear interpolation. There are also efficient implementations of other types of interpolation [14]. The procedure consists of two parts (see Fig. 1d). First we calculate linear interpolation coefficients, i.e. for each t in the union, we search for $t_i < t < t_{i+1}$ and calculate $t_{i+1} - t_0$. Since the time series are non-uniformly sampled this operation uses vectorized search (upper_bound from *Thrust*). The second step uses a linear interpolation GPU hardware support and uses previously computed factors.

Aggregation: Aggregation works on equally sampled time series. For each time point we calculate aggregation across data values in all time series. Each thread in a loop analyses the values of all time series for a single point in time. Then it writes aggregated value to global memory. See Fig. 1e for overview.

3 Prototype Query processing

Query Processing: The experiments were carried out using a common query for monitoring systems: *Calculate an aggregation of all the time series for a given metric for a specified period of time.* It is a general task which is a starting point for many other analytical methods like finding extreme values, pattern matching or other data mining algorithms. It covers all important aspects of query processing: data retrieval, combination of many time series, missing data and data aggregation.

Data Settings: A synthetic set of time series for a single metric with different tags was prepared. It may be treated as one parameter measurement on a set of distributed sensors. The simulated measurements correspond to 600 time series with measurement every 300 seconds with random 10% of elements missing in each time series, which gives approximately $600 \times 16.1K \approx 9.6M$ data points. Additionally, the synthetic data has been prepared to obtain different compression ratios seen in real applications [13].

Environment Settings: Experiments were carried out on one instance of HBase, query processing was conducted on the database server. Hardware configuration: Two six core processors *Intel® Xeon® E5649 2.53GHz*, 8GB RAM and *Nvidia® Tesla M2070* card. Tests were carried out using 600 time-series containing from 2.0K to 16.1K of observations, average processing time for 25 launches was taken. Because processed data in most cases fit in the HBase cache, configuration with LZO (Lempel-Ziv-Oberhumer) compression achieves only slightly better results than with no compression. In industrial applications, queries are less likely to hit the cache and the acceleration of LZO compression is higher.

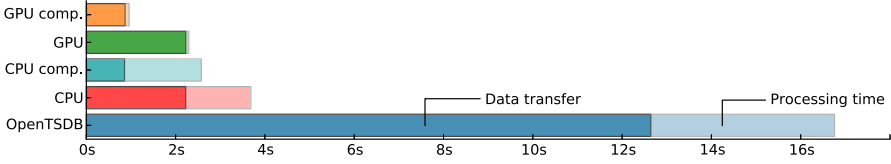
OpenTSDB: A modified version of a console client (modified to log query execution time after doing a warm-up phase of Java virtual machine) and Hbase configured with LZO compression were used in experiments.

Prototype: Developed using C++ and CUDA C++ and Thrift protocol. HBase was configured without compression, instead highly tuned lightweight (de)compression algorithms for time series were used.

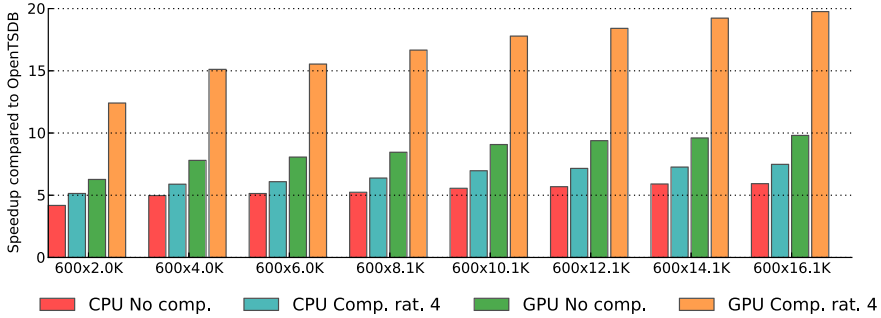
3.1 Results and Discussion

The comparison of OpenTSDB and our prototype performance is eligible due to similar architecture of both solutions and identical query processing work-flow. All differences are discussed below. The following factors were considered: the data transfer time (the time between sending a query to HBase and receiving the results) as well as the time needed to process the data (including data decompression), the time required to exchange data with the GPU (if used) and the processing time.

A detailed timeline for query execution with fixed data size (600 time series \times 16.1K observations) is provided in Figure 2a. We can observe that processing in OpenTSDB



(a) Execution time 600 time-series, 16.1K observation each (lower is better)



(b) Prototype speedup compared to OpenTSDB (higher is better)

Fig. 2. Measured results

takes only 25% of query time and despite being 2.8 times slower than CPU prototype, it is not the main bottleneck. Better performance is not just a matter of changing Java to C++. It is *data compaction* to reduce processed data size and number of fetched rows and columns and increase efficiency of data transfer. Using data compaction, data transfer performance significantly increases (5.7 times faster) for both prototypes (CPU and GPU). But still most of the time is spent on communication with the database. What is more GPU is 21x faster than CPU when comparing data processing speed. Thus calculations are limited by the data transfer. It is therefore necessary to improve data transfer in order to achieve better results. This is done by using efficient lightweight compression implementation [12]. Lightweight compression introduces only a slight improvement in CPU prototype. This is because of the relatively long time needed for the data processing (almost $\frac{1}{4}$ of total time – see *CPU* in Fig. 2a). This is because the lightweight compression significantly reduces data transfer time, but it also increases the data processing time (see *CPU comp.* in Fig. 2a). Computation and communication with the GPU is only a small fraction of the entire query and decompression adds only a small overhead to the query (see *GPU* and *GPU comp.* in Fig. 2a).

Figure 2b presents the resulting acceleration obtained on CPU and GPU prototype (in comparison to OpenTSDB query) on different data sizes. Both figures include CPU and GPU prototypes with and without lightweight compression. Due to the page limit only results for one compression ratio (4) are presented. Notice that the size of the data is important and better results can be obtained on larger data sets. It is also worth noting that the data transfer is often a bottleneck in many GPGPU applications. This was also the case, however, through the use of a lightweight compression, data transfer is highly improved, thereby significantly speeding up the execution of the query.

4 Conclusions and Future Work

Time series databases play a crucial role in many branches of industry. Machine generated measurements require fast, real-time insertion and almost real-time querying. We showed that in case of computations dedicated to time series the existing solutions may be improved by utilization of GPU processors. So far data intensive application had to overcome the problem of additional CPU to GPU data transfer cost. Only algorithms of more than linear computation time complexity could benefit from parallel GPU processing. In this paper we showed that by introduction of fine tuned compression methods we can improve these results. Especially time series processing may speed-up significantly when compared to industrial solutions or experimental CPU prototypes.

Our future work will concentrate on query optimization in hybrid CPU/GPU environment, query execution on partially compressed data and on developing dynamic compression planer.

References

1. Apache HBase (2013), <http://hbase.apache.org>
2. Business Intelligence and Analytics Software - SAS (2013), <http://www.sas.com/>
3. Jedox - website (2013), <https://www.jedox.com>
4. OpenTSDB - A Distributed, Scalable Monitoring System (2013), <http://opentsdb.net/>
5. ParStream - website (2013), <https://www.parstream.com>
6. TempoDB – Hosted time series database service (2013), <https://tempo-db.com/>
7. The R Project for Statistical Computing (2013), <http://www.r-project.org/>
8. Chang, F., et al.: Bigtable: A Distributed Storage System for Structured Data. In: OSDI 2006: Seventh Symposium on Operating System Design and Implementation, pp. 205–218 (2006)
9. Cloudkick. 4 months with cassandra, a love story (March 2010), <https://www.cloudkick.com/blog/2010/mar/02/4-months-with-cassandra/>
10. Fang, W., He, B., Luo, Q.: Database compression on graphics processors. Proceedings of the VLDB Endowment 3(1-2), 670–680 (2010)
11. ParStream. ParStream - Turning Data Into Knowledge - White Paper. Technical report (2010)
12. Przymus, P., Kaczmarek, K.: Improving efficiency of data intensive applications on GPU using lightweight compression. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM-WS 2012. LNCS, vol. 7567, pp. 3–12. Springer, Heidelberg (2012)
13. Przymus, P., Rykaczewski, K., Wiśniewski, R.: Application of wavelets and kernel methods to detection and extraction of behaviours of freshwater mussels. In: Kim, T.-h., Adeli, H., Slezak, D., Sandnes, F.E., Song, X., Chung, K.-i., Arnett, K.P. (eds.) FGIT 2011. LNCS, vol. 7105, pp. 43–54. Springer, Heidelberg (2011)
14. Ruijters, D., ter Haar Romeny, B.M., Suetens, P.: Efficient gpu-based texture interpolation using uniform b-splines. Journal of Graphics, GPU, and Game Tools 13(4), 61–69 (2008)
15. Unde, P., et al.: Architecting the database access for a it infrastructure and data center monitoring tool. In: ICDE Workshops, pp. 351–354. IEEE Computer Society (2012)
16. Wu, L., Storus, M., Cross, D.: Cs315a: Final project cuda wuda shuda: Cuda compression project. Technical report, Stanford University (March 2009)
17. Yan, H., Ding, S., Suel, T.: Inverted index compression and query processing with optimized document ordering. In: Proc. of the 18th Intern. Conf. on World Wide Web, pp. 401–410. ACM (2009)
18. Zukowski, M., Heman, S., Nes, N., Boncz, P.: Super-scalar ram-cpu cache compression. In: ICDE 2006, Proc. of the 22nd intern. conf. on Data Engineering, pp. 59–59. IEEE (2006)

Rule-Based Multi-dialect Infrastructure for Conceptual Problem Solving over Heterogeneous Distributed Information Resources*

Leonid Kalinichenko, Sergey Stupnikov, Alexey Vovchenko,
and Dmitry Kovalev

Institute of Informatics Problems, Russian Academy of Sciences, Moscow, Russia
{leonidandk,itsnein,dm.kovalev}@gmail.com, ssa@ipi.ac.ru

Abstract. An approach for applying a combination of the semantically different rule-based languages for interoperable conceptual programming over various rule-based systems (RS) and relying on the logic program transformation technique recommended by the W3C Rule Interchange Format (RIF) is presented. Such approach is coherently combined with the heterogeneous data base integration applying semantic rule mediation. The basic functions of the infrastructure implementing the multi-dialect conceptual specifications by the interoperable RS and mediator programs are defined. The references to the detailed description of the infrastructure application for solving complex combinatorial problem are given. The research results show the usability of the approach and of the infrastructure for declarative, resource independent and re-usable data analysis in various application domains.

Keywords: conceptual specification, RIF, logic rule languages, database integration, mediators, BLD, CASPD, multi-dialect infrastructure, rule delegation.

1 Introduction

The paper¹ investigates a novel methodology and infrastructure supporting conceptually-driven problems specification and solving. This research is motivated by aiming at the specifications reusability in various applications over different sets of data, widely diverse data and knowledge semantic integration capability, and for accumulation of reproducible data analysis and problem solving methods and experience in various application domains.

The objective of the work is to expose the current practically reachable limit of declarative conceptual specification construction applying the wealth of various available facilities of logic programming, knowledge representation, semantic Web and heterogeneous database mediation in a coherent, cooperative way.

* This research has been done under the support of the RFBR (project 11-07-00402-) and the Program for Basic Research of the Presidium of RAS.

¹ Due to the size limitations the paper is to be considered as an *extended abstract*.

Specifically the approach proposed is aimed at conceptual modeling of data intensive domains (DID) in rule-based declarative languages possessing different, complementary semantics and capabilities combined with the methods for heterogeneous data mediation and integration. Besides that, the approach might also be applicable for the programming and composition of complex analytical pipelines in an under-standable form applying appropriate high-level languages to express the analytics intended for inferring knowledge from data [1].

In the work presented the issues of interoperability and integration of various information resources (such as data and knowledge bases, software services, ontologies) for the problem solving are investigated on the basis of two approaches: (1) constructing of the unifying extensible language providing for semantic preserving representation in it of various information resource (IR) languages; (2) creation of the unified extensible family of rule-based languages (dialects) and a model of interoperability of the programs in such dialects. The first approach is based on the SYNTHESIS language [2][3] accompanied by methods and facilities for constructing of its extensions and the canonical information model (CIM). CIM is used for development of subject mediators positioned between the users, conceptually formulating problems in terms of the GLAV-based mediator, and various distributed IRs (such as databases and services) needed for a specific application.

Another, multi-dialect approach for IR interoperability applied in the current work is based on the RIF (Rule Interchange Format) recommendation [4] of W3C. RIF introduces a unified rule-based language (dialect) family together with a methodology for constructing of semantic preserving mappings in such dialects of specific languages used in various Rule-based Systems (RS). For inclusion of a rule-based language of a specific RS into a set of interoperable dialects it is required to develop for this RS two semantic preserving transformers — from the RS language into a RIF dialect (a *supplier* role) and from the dialect into the RS language (a *consumer* role). Every RIF dialect can have its own semantics different from other dialects. For the present RIF the recommendations are defined for the very basic dialects. E.g., the RIF-BLD (Basic Logic Dialect [5]) corresponds to the Horn logic with some extensions. Examples of its subdialects that still are expected to be accepted as the W3C recommendations include the RIF-CLPWD (Core Logic Programming Well-founded Dialect), which uses well-founded semantics (WFS) with the default negation and functions, and RIF-CASPD (Core Answer Set Programming Dialect [6]) which uses answer set programming semantics (ASP), known also as the stable model semantics. WFS and ASP can be used for different purposes. ASP-based systems are specifically oriented on solving of complex combinatorial (NP-complete) problems whereas WFS-based systems are computationally complete and can be used as the general purpose logic programming facilities. RIF recommendations have also defined the necessary concepts to ensure compatibility of RIF with RDF and OWL , in spite of dissimilarity of their syntaxes and semantics.

The paper annotates the results obtained including the description of an approach and an infrastructure supporting (a) the application domain conceptual

specification and problem solving algorithms definitions based on the combination of the heterogeneous database mediation technique and rule-based multi-dialect facilities; (b) interoperability of distributed multi-dialect rule-based programs and mediators integrating heterogeneous databases; (c) rule delegation approach in the multi-dialect environment. The infrastructure based on the SYNTHESIS environment and RIF standards has been implemented. The approach for multi-dialect conceptualization of a problem domain, rule delegation and rule-based programs and mediators interoperability has been demonstrated on a real NP-complete application in the finance domain. For the conceptual definition of the problem we use OWL for the domain concepts definition and programs in two dialects — RIF-BLD mapped into the SYNTHESIS mediator program and RIF-CASPD mapped into ASP-based DLV [7] program.

The paper is structured as follows. After the introduction, the section containing an overview of the approach and an infrastructure for distributed multi-dialect rule-based programs support is given. In the third section the references to the detailed descriptions of the application example are provided (they could not be included due to the limit imposed on the paper size). A related work section and the conclusion which summarizes the results and outlines plans for the future work close the paper.

2 Infrastructure of the Multi-dialect Environment for Distributed Rule-Based Programs Interoperability

2.1 Conceptual Programming and Conceptual Schema

The aim of the novel infrastructure proposed is a conceptual programming of problems in RIF dialects and an implementation of conceptual programs using declarative languages of the RSs. These languages possess different capabilities and semantics and provide for programming over heterogeneous resources of data, programs and ontologies. Access to the resources is provided by concrete RSs or subject mediators. Conceptual multi-dialect logic programs specify the algorithms for problem solving in a subject domain. They are implemented using their transformation into a RS or a mediator programs.

Conceptual schema of a problem (class of problems) is defined in the frame of a subject domain and consists of a set of *RIF-documents* (document is a specification unit of RIF). Every document contains groups of rules. The subject domain conceptualization is performed using OWL 2 ontologies containing entities of the domain and their relationships (Fig. 1, right-hand part). Ontologies constitute the *conceptual specification* of the domain. Names of the entities (classes and attributes) are used in the rules of the RIF-documents. Ontologies are imported into RIF-documents specifying an import profile, for instance, *OWL Direct*. A profile defines a semantics of an OWL ontology.

Modular construction of the conceptual schema is based on the techniques of document import and link. Documents import other documents having the same

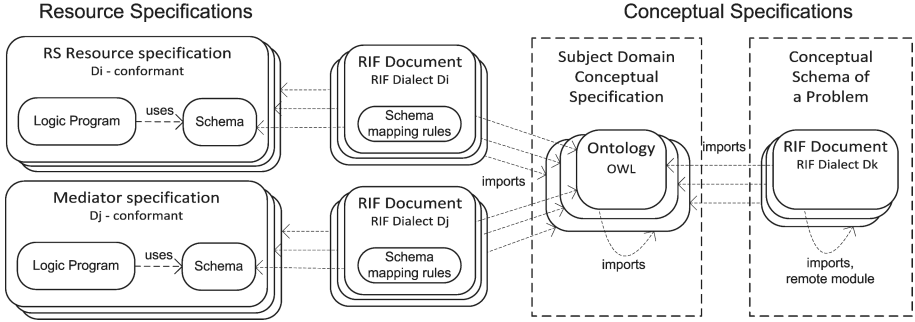


Fig. 1. Conceptual schema and resource specifications

semantics (the *Import* directive) or link documents defined using other dialects and having different semantics (remote module directive *Module*).

Retrieving results of problem solving is performed by querying the conceptual schema of a problem. A query is formulated over a RIF-document of a schema using the dialect of the document.

2.2 Resources Relevance to a Problem and Mapping of their Schemas into the Conceptual Specification

In the proposed infrastructure (1) the *logic programs* implementing RIF-documents of the conceptual schema in specific RSs and (2) the *subject mediators* supporting collections of facts as the result of heterogeneous databases integration are considered as resources.

A schema S_R of a resource R is a set of entities (classes or relations and their attributes) corresponding to extensional and intensional predicates of the resources. The R_S of every resource R should be a conformant D_R consumer, where D_R is a RIF dialect (Fig. 1, left-hand part). Conformance is formally defined using formula entailment and language mappings.

The resource R is relevant to a RIF-document d of a conceptual schema if:

- D_R is a subdialect of the document d dialect and
- entities of schema S_R (if they exist) are *ontologically relevant*² to entities of the subject domain conceptual specification the names of which are used in d for extensional predicates.

The schema of a relevant resource is mapped into the subject domain specification. This means that conceptual entities referenced in the document d are expressed in terms of entities of the schema S_R using logic rules of the D_R dialect. These rules constitute separate RIF-document (Fig. 1, middle part).

² In this paper we do not consider methods for schema ontological matching.

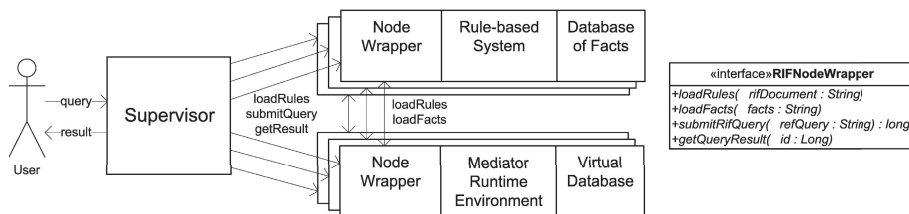


Fig. 2. Peer-to-peer multi-dialect network architecture

2.3 Implementation of the Conceptual Schema Programs

Programs of the conceptual schema are implemented in P2P environment formed by relevant resources which are related to conceptual specification by mapping rules (Fig. 1, middle part).

Resources are peers (nodes) of the P2P environment. Peers communicate using a technique for distributed execution of the logic programs. The basic notion of the technique is *delegation* – transferring facts and rules from one peer to another. A peer is a combination of a wrapper, a RS resource or a mediator, a logic program and possibly a collection of facts (Fig. 2).

A wrapper transforms programs and facts from the specific RIF dialect into the language of the RS or mediator and vice versa. A wrapper also implements the delegation mechanism. A definition of the delegation is given in the latter part of this section. Transferring facts and rules among peers is performed using RIF dialects. Wrappers implement an interface *RIFNodeWrapper* (Fig. 2).

A special component (*Supervisor*) of the architecture stores shared information of the environment, i.e. domain conceptual specification and conceptual schema of the problem, a list of the relevant resources, RIF-documents combining logic rules for the conceptual specification and a resource schema mapping.

Implementation of the conceptual schema includes the following steps:

1. Rewriting of the conceptual schema into the RIF-programs of resources performed by the *Supervisor*. A rewriting includes (1) replacing the document identifiers (used to mark predicates) by peer identifiers and (2) adding schema mapping rules to programs (Fig. 1, middle part).
2. A transfer of the rewritten programs to peers containing resources relevant to the respective conceptual documents.
3. A transformation of the RIF-programs into the concrete RS languages.
4. Execution of the produced programs in peers.

During the process of rewriting of the conceptual schema into the resource programs a structure of a real P2P network is formed. A virtual node corresponding to a RIF-document of the conceptual schema is replaced by one or more peers corresponding to resources relevant to the document. Relationships between virtual nodes defined by remote or imported terms are replaced by relationships between real peers also defined by remote or imported terms. To

implement remote and imported terms a rule delegation mechanism similar to one proposed in WebdamLog [8] is used (for more details look at the *Related Work* section).

In the general case, programs transferring to some peer may include *nonlocal* rules. These rules contain remote or imported terms. On the contrary, *local* rules do not contain such terms. For simplicity only remote terms are mentioned in the latter part of this section. To make possible an execution of a program in a peer the program should be *normalized*, i.e. transformed into an equivalent program including only local rules and *delegation rules*. Delegation rules produced during normalization are transferred to the respective peers. Normalized programs accompanied by delegated rules are executed in peers. To save space the details of normalization and the algorithm of program execution are omitted here. It can be found at http://synthesis.ipi.ac.ru/synthesis/projects/RuleInt/Conceptual_Schema_Programs.htm Web-site.

3 The Use-Case for the Multi-dialect Infrastructure

The capabilities of the multi-dialect architecture are illustrated with the solution of the *investment portfolio diversification problem* [9]. The portfolio is a collection of securities (such as equities or bonds), and its size is the number of securities in the portfolio. The task is to build a diversified portfolio of maximum size. Diversification means that the prices of the securities in portfolio are almost independent of each other. If the price of one security falls, it will not significantly affect the prices of other. Thus the risk of a portfolio sharp decrease is significantly reduced.

The input data to the problem is a set of securities and corresponding time series (such as closing price) for each security. Also the predetermined price correlation value is specified. It serves as maximum risk measure of a sharp reduction of the portfolio value. The output is the maximum size subset of securities, for which the pairwise correlation will be less than the specified one (the *Pearson correlation* is used).

The problem is divided into the following tasks: (1) computation of the security pairwise correlations (for specified dates) and (2) calculation of the maximum satisfying subset of securities.

To solve the first task the financial services *Google Finance*³ and *Yahoo! Finance*⁴ are considered, both of which provide current and historical information about stock prices, currencies, bonds, stock indexes, etc. Mediator environment is used to solve the problem of resource integration [3].

Second task is formulated as follows. Let G be a graph where vertices are securities, and an edge between two securities exists if absolute value of their correlation is less than a specified number. So, this is a well-known NP-complete problem – finding a maximum clique in an undirected graph. ASP logic programming systems, e.g. DLV [7], are well-suited for solving such problems.

³ <https://www.google.com/finance>

⁴ <http://finance.yahoo.com/>

Application domain conceptual specification (ontology) of security historical prices is defined using OWL [10]. The conceptual schema of the problem includes two documents that correspond to the specified tasks. The first of the documents contains a program that calculates the correlation graph of securities based on the prices in a given period of time. The document is defined in the RIF-BLD dialect [5]. The second document contains a program that computes the maximum clique in a graph of correlations. The document is defined in RIF-CASPD dialect [6].

Resources for the problem of the investment portfolio diversification are subject mediator, in which the *Google Finance* and the *Yahoo! Finance* services are integrated, and a program in rule-based programming system DLV [7]. Due to the limited space, specifications and detailed description of portfolio diversification problem solving as well as the results obtained are omitted here. They can be found at http://synthesis.ipi.ac.ru/synthesis/projects/RuleInt/Conceptual_Use_Case_Example.htm Web-site. A test execution of the use-case programs in the developed infrastructure for *Standard & Poor's 500* securities during 2012 gave 11 maximal portfolios of size 10 each.

4 Related Work

The rule exchange using RIF dialect for production rule systems (RIF-PRD) is discussed in [13] [14]. In such case the production rule systems share the same operational semantics opposed to our approach studying the problem of rule exchange between systems with different semantics.

Several approaches intended for specifying declarative distributed programs and managing data in distributed environment exist [8][11][12]. In contrast to multi-dialect approach, a single declarative language is used in each of the proposed systems. Usually it is a conventional Datalog extended with the notion of localization and possibly other non-datalog constructs [12]. In the multi-dialect approach location is specified with RIF remote and imported terms.

Conceptual notion of *delegation* applied in our approach coincides with the notion of delegation in Webdamlog defined as “the possibility of installing a rule at another peer. In its simplest form, delegation is essentially a remote materialized view. In its general form, it allows peers to exchange rules, i.e., knowledge beyond simple facts, and thereby provides the means for a peer to delegate work to other peers” [8]. Actually, current implementation supports a remote materialized view. Extending the approach for delegation of knowledge is a future work. The idea of program normalization is similar to the rule localization rewriting step described in [12].

5 Conclusion

The approach presented is the first attempt of introducing the multi-dialect interoperable conceptual programming over various semantically different rule-based programming systems relying on the logic program transformation technique recommended by W3C RIF. We show also how to coherently combine

such approach with the heterogeneous data bases integration applying the semantic mediation. The results obtained so far are quite encouraging for future work aimed at reaching of the conceptual specifications reusability in various applications over different sets of data, as well as for sharing and accumulation of reproducible data analysis and problem solving methods and experience in various data intensive domains.

References

1. Challenges and Opportunities with Big Data. A community white paper developed by leading researchers across the United States (2012), <http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf>
2. Kalinichenko, L.A., Stupnikov, S.A., Martynov, D.O.: SYNTHESIS: A language for canonical information modeling and mediator definition for problem solving in heterogeneous information resource environments. M.: IPIRAN, 171 p (2007)
3. Kalinichenko, L.A., Briukhov, D.O., Martynov, D.O., Skvortsov, N.A., Stupnikov, S.A.: Mediation Framework for Enterprise Information System Infrastructures. In: Proc. of the 9th International Conference on Enterprise Information Systems, ICEIS 2007, Funchal. Databases and Information Systems Integration, pp. 246–251 (2007)
4. Boley, H., Kifer, M. (eds.): RIF Overview. W3C Working Group Note, 2nd edn. (February 5, 2013)
5. Boley, H., Kifer, M.(eds.): RIF Basic Logic Dialect. W3C Recommendation, 2nd edn. (February 5, 2013)
6. Heymans, S., Kifer, M.(eds.): RIF Core Answer Set Programming Dialect (2009), <http://ruleml.org/rif/RIF-CASPD.html>
7. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV System for Knowledge Representation and Reasoning. *ACM Transactions on Computational Logic* 7(3), 499–562 (2006)
8. Abiteboul, S., Bienvenu, M., Galland, A., et al.: A rule-based language for Web data management. In: Proceedings 30th ACM Symposium on Principles of Database Systems, pp. 283–292. ACM Press (2011)
9. Sharpe, W., Alexander, G.J., Bailey, J.W.: Investments. Prentice Hall (1998)
10. Bock, C., et al. (eds.): OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation, 2nd edn. (December 11, 2012)
11. Grumbach, S., Wang, F.: Netlog, a rule-based language for distributed programming. In: Carro, M., Peña, R. (eds.) PADL 2010. LNCS, vol. 5937, pp. 88–103. Springer, Heidelberg (2010)
12. Loo, B.T., et al.: Declarative networking: language, execution and optimization. In: ACM SIGMOD Conference Proceedings, pp. 97–108 (2006)
13. Cosentino, V., Del Fabro, M.D., El Ghali, A.: A model driven approach for bridging ILOG Rule Language and RIF. In: Proceedings of the 6th International Symposium on Rules, RuleML 2012. CEUR-WS.org, vol. 874, pp. 96–102 (2012)
14. Gonzalez-Moriyon, G.: Final steel industry public demonstrators. EU-IST Integrated Project 2009-231875 ONTORULE D5.5 Report (2012)

Distributed Processing of XPath Queries Using MapReduce^{*}

Matthew Damigos¹, Manolis Gergatsoulis¹, and Stathis Plitsos²

¹ Database and Information Systems Group (DBIS),
Department of Archives and Library Science, Ionian University, Corfu, Greece
mgdamig@gmail.com, manolis@ionio.gr

² Department of Management Science and Technology,
Athens University of Economics and Business, Athens, Greece
stathisp@aueb.gr

Abstract. In this paper we investigate the problem of efficiently evaluating XPath queries over large XML data stored in a distributed manner. We propose a MapReduce algorithm based on a query decomposition which computes all expected answers in one MapReduce step. The algorithm can be applied over large XML data which is given either as a single distributed document or as a collection of small XML documents.

1 Introduction

XML is a widespread format used for exchanging information on the Web, and, in general, for representing semi-structured data. The efficient querying and analysing large amount of web data is now being broadly recognized as a significant challenge in many areas, such as system designing, data analysis, decision-making, marketing and biology research. The management of the information appearing in a collection of XML data is achieved by using XML administrative languages such as XPath, XSLT and XQuery [6]. In this paper, we focus on XPath, which constitutes the basis for most of the other XML administrative languages. Furthermore, we use the MapReduce distributed framework [4] to process and manage large amount of XML data. MapReduce is widely used for processing large amount of data using a cluster of commodity machines. Boasting a simple, fault-tolerant and scalable paradigm, it has established itself as dominant in the area of massive data analysis.

In this paper we investigate the problem of evaluating XPath queries over large XML data which is stored in a cluster of commodity machines. The XML query evaluation in the MapReduce framework has been little investigated in the past. Query decomposition to sub-queries depending on the accessibility of data distributed to a fixed number of sites has been exploited in [7]. However, this

^{*} This research was supported by the project “Handling Uncertainty in Data Intensive Applications”, co-financed by the European Union (European Social Fund - ESF) and Greek national funds, through the Operational Program ”Education and Lifelong Learning”, under the research funding program THALES.

approach is agnostic to data distribution thus adhering to the distributed file system paradigm. Partial evaluation of XPath queries over a distributed XML document is also the subject of [3]. However, this approach assumes a sole coordinator entrusted with the joining of the partial results. In addition, the authors also propose a MapReduce algorithm which evaluates boolean queries. Note that this algorithm uses a single reduce task to compute the final answer. The problem of evaluating twig pattern queries on distributed XML data is investigated in [2], where the authors propose a system which computes the result based on indexing. In [5], MRQL, a query language over MapReduce, is introduced for the analysis of XML data. Finally, [9] explores the idea of a parallelized processing workflow of XML fragments in a MapReduce environment.

The main contribution of this paper is that it proposes a MapReduce algorithm, called HoX-MaRe, which computes all expected answers in one MapReduce step (Section 3.1). Our algorithm is based on query decomposition and a horizontal fragmentation method for the XML document and ensures that it computes all answers that would be resulted when the query is evaluated in a single machine. Preliminary experimental results concerning the performance of our algorithm are presented in Section 3.2.

2 Preliminaries

In this section we present the preliminary definitions of the concepts used in the subsequent sections. Consider a directed, rooted, labelled tree t (*tree* for short), where its labels come from an infinite set Σ . We denote $\mathcal{N}(t)$ and $\mathcal{E}(t)$ the set of nodes and edges, respectively, of t , and we write $label(n)$ to denote the label of a node n of t . We refer to the unique path through which n is reachable from root of t (denoted by $root(t)$) as *reachable path* of a node n in $\mathcal{N}(t)$. If there is an edge (n_1, n_2) in $\mathcal{E}(t)$, the node n_2 is a *child* of n_1 . A node n'_2 of t is a *descendant* of a node n'_1 of t if t has a path from n'_1 to n'_2 . A *branching node* is each node in $\mathcal{N}(t)$ having at least two children.

We consider two types of trees that represent XML documents and queries in XPath, respectively. An XML document is represented by a tree (also called *XML tree*) having text, or numbers, associated with leaf-nodes; while the XPath queries are different from XML trees in four aspects. First, the labels of a query come from the set $\Sigma \cup \{*\}$, where $*$ is the “wildcard” symbol. Second, a query P has two types of edges: $\mathcal{E}_/ (P)$ is the set of child edges (represented by a single line) and $\mathcal{E}_{//} (P)$ is the set of descendant edges (represented by a double line). Third, a query P has an output node (or output, for short), denoted by $out(P)$, and is represented by a circled node. Fourth, each leaf-node which is not the output node may be associated with a condition; instead of text and numbers that are associated with leaf-nodes of an XML tree. The *selection path* of a non-boolean query Q is the path from the root to the output node. A *subquery* of Q is an XPath query having a subset of both the nodes and the edges of Q .

The result of applying a query Q on an XML tree t is based on a set of mappings from the nodes of Q to the nodes of t , called embeddings. An

embedding from Q to t is a mapping $e : \mathcal{N}(Q) \rightarrow \mathcal{N}(t)$ with the following properties: (1) Root preserving: $e(\text{root}(Q)) = \text{root}(t)$, (2) Label preserving: For all nodes $n \in \mathcal{N}(Q)$, either $\text{label}(n) = *$ or $\text{label}(n) = \text{label}(e(n))$, (3) Child preserving: For all edges $(n_1, n_2) \in \mathcal{E}_/(Q)$, we have that $(e(n_1), e(n_2)) \in \mathcal{E}(t)$, (4) Descendant preserving: For all edges $(n_1, n_2) \in \mathcal{E}_{//}(Q)$, the node $e(n_2)$ is a proper descendant of the node $e(n_1)$, and Leaf preserving: For each leaf-node $n_\ell \in \mathcal{N}(Q)$ associated with a condition either of the form “ $\text{text}() = \text{str}$ ” or of the form “ $\text{val}() \text{ op num}$ ” we have that either the text associated with $e(n_\ell)$ is identical to str or the number C associated with $e(n_\ell)$ satisfies the condition “ $C \text{ op num}$ ”, respectively. We recall that *op* stands for one of the arithmetic comparison operators $=, \neq, <, >, \geq, \leq$, and *num* is a number. The result $Q(t)$, now, of applying a non-boolean query Q on a tree t is formally defined as follows: $Q(t) = \{e(\text{out}(Q)) \mid e \text{ is an embedding from } Q \text{ to } t\}$. If Q is a boolean query then the result $Q(t)$ is “*true*”, only if there is an embedding from Q to t .

2.1 MapReduce Framework

The MapReduce is the programming model for processing large datasets in a distributed manner. The storage layer for the MapReduce framework is a Distributed File System (DFS), such as the Hadoop Distributed File System (HDFS), and is characterized by the block size which is typically 16-128MB in most of DFSs. Creating a MapReduce job is straightforward. The user defines two functions, the *Map* and the *Reduce* function, which run in each cluster node, in isolation. The map function is applied on one or more files, in DFS, and results $\langle \text{key}, \text{value} \rangle$ pairs. This process is called *Map task*. The nodes that run the Map tasks are called *Mappers*. The *master controller* is responsible to route the pairs to the *Reducers* (i.e., the nodes that apply the reduce function on the pairs) such that all pairs with the same key initialize a single reduce process, called *reduce task*. The reduce tasks apply the reduce function in the input pairs and also result $\langle \text{key}, \text{value} \rangle$ pairs. This procedure describes a *MapReduce step*. Furthermore, the output of the reducer can be set as the input of a map function, which gives to the user the flexibility to create procedures of multiple steps.

2.2 Fragmentations of XML Data

Considering an arbitrary method of attaching ids to element-nodes of an XML tree t we define, in this section, a fragmentation of an XML tree which preserves the structure of the initial tree. We say that a set \mathcal{T} of XML trees (called *fragments*) forms a *horizontal fragmentation* of t if for each fragment \mathcal{F} in \mathcal{T} the following hold. (1) For each node $n_{\mathcal{F}}$ of \mathcal{F} , there is a node n of t having the same reachable path to that of $n_{\mathcal{F}}$, (2) for each node n of t , there is a node $n_{\mathcal{F}}$ of a document \mathcal{F} in \mathcal{T} having the same reachable path to that of n , and (3) each fragment in \mathcal{T} contains at least one leaf-node of t , which is not contained in other fragment in \mathcal{T} . For example, in Fig. 1, the XML trees \mathcal{F}_1 and \mathcal{F}_2 represent the fragments of a horizontal fragmentation of the XML tree t . To verify this, notice that \mathcal{F}_1 and \mathcal{F}_2 are obtained by splitting t at the node m_7 , and the path

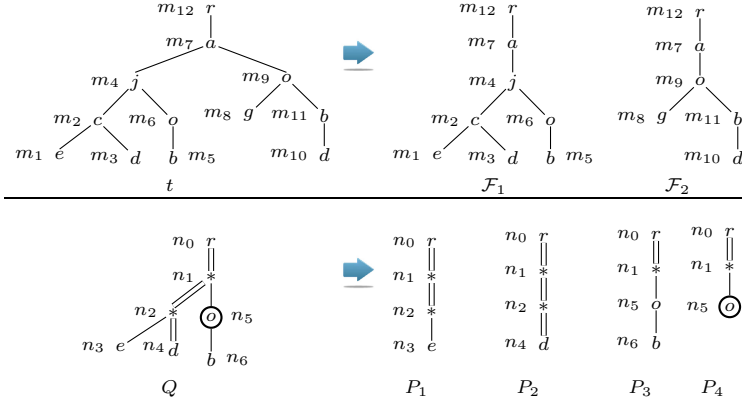


Fig. 1. (a)XML fragmentation, (b) XPath Query Decomposition

from the root of t to m_7 is included in both fragments. It is easy to see that the nodes of t keep their ids after the fragmentation. In this way the child-parent relationship, as well as the structure of t , are preserved in each fragment.

The horizontal fragmentation can be used to partition the information of an XML tree to several XML fragments. In the following, we consider that the size of each fragment does not exceed the maximum block size.

3 XPath Evaluation on MapReduce Framework

The problem of efficiently evaluating XPath queries over a large amount of XML data using the MapReduce framework is formally stated as follows. Considering a distributed collection \mathcal{T} of XML trees which form a horizontal fragmentation of a large XML tree t and an XPath query Q , we want to compute the answers that would be resulted by $Q(t)$, in parallel, using the MapReduce framework. In Section 3.1, we propose an algorithm, called *HoX – MaRe Algorithm*, which deals with this problem in one MapReduce step.

Consider the horizontal fragmentation of t illustrated in Fig. 1. It is easy to verify that evaluating Q over each fragment (\mathcal{F}_1 and \mathcal{F}_2), in isolation, using one of the conventional methods of XPath evaluation, we get only the node m_6 ; i.e., we miss the node m_9 . This node, however, should be included in $Q(t)$. Intuitively, the reason why the node m_9 is not resulted from the evaluation of Q on the fragments of t , is that the XML data needed to obtain the embedding giving m_9 is split in the two different fragments. To deal with this problem we define the concept of *query decomposition*. Let D_Q be the set of XPath queries obtained from an XPath query Q as follows. For each leaf node n of Q which is not output we add the reachable path of n to D_Q . Then we also add to D_Q the query consisting of the selection path of Q ; which is the only non-boolean query in D_Q . The set D_Q gives the *decomposition* of Q .

Example 1. The XPath query Q illustrated in Fig. 1 has 3 leaf nodes; the n_3 , n_4 and n_6 . From these nodes we obtain the queries P_1 , P_2 and P_3 , respectively, and add them to a set D_Q . Notice that each query is given by the reachable path of each leaf node. In addition, we add to D_Q the selection path of Q , which is given by the query P_4 . The set D_Q gives the decomposition of Q .

The following theorem describes how we can find an embedding from a XPath query Q to a tree t when we have already found a set of embeddings from the queries in the decomposition of Q to t .

Theorem 1. *Let Q be an XPath query in $\mathcal{XP}^{\{*,[,],//,\wedge\}}$, t be an XML tree and $D_Q = \{P_1, \dots, P_n\}$ be the decomposition of Q such that P_n is the selection path of Q . Then for each node $o \in \mathcal{N}(t)$ the following are equivalent:*

- (1) *There is a set of embeddings $M = \{e_1, \dots, e_n\}$ such that (a) for each i , with $1 \leq i \leq n$, e_i is an embedding from P_i to t , (b) for each $n \in \mathcal{N}(Q)$ there aren't two embedding $e_i, e_j \in M$ such that $e_i(n) \neq e_j(n)$, and (c) $e_n(\text{out}(P_n)) = o$.*
- (2) *There is an embedding e from Q to t such that $e(\text{out}(Q)) = o$.*

The Condition 1 in Theorem 1 can be easily extended to cover the case that each query $R_i \in D_Q$ maps on a fragment in a horizontal fragmentation \mathcal{T} of t (instead of a mapping from P_i directly to t).

Corollary 1. *If we replace the Condition 1(a) in Theorem 1 with “(a’) for each i , with $1 \leq i \leq n$, e_i is an embedding from P_i to a fragment \mathcal{F} , where \mathcal{F} is contained in a horizontal fragmentation \mathcal{T} of t ”, then Theorem 1 still holds.*

Corollary 1 implies a two-steps method for computing all nodes in $Q(t)$ in a distributed manner. Particularly, we firstly compute, in parallel, the embeddings from each query in D_Q to each fragment of \mathcal{T} , then these embeddings are emitted, along with the answers of the non-boolean queries, and in the second phase, the embeddings are combined properly in order to give the output nodes in $Q(t)$.

3.1 HoX-MaRe Algorithm

In this section, we present a MapReduce algorithm, called *HoX – MaRe* Algorithm, which computes the answer of an XPath query Q when Q is posed on a distributed XML tree given by a horizontal fragmentation \mathcal{T} . The fragments in \mathcal{T} are stored in a DFS. In the following we consider that the branching nodes of Q are mapped, using a bijection h , on an integer between 1 and $|\mathcal{N}_B(Q)|$, where $|\mathcal{N}_B(Q)|$ is the number of branching nodes of Q . In addition, we suppose that h has initially been sent to all mappers. The Map and the Reduce function of the algorithm are formally depicted in Fig. 2.

Map function: The Map function gets as input a fragment \mathcal{F} in \mathcal{T} and performs the following operations. Initially, the decomposition D_Q of Q is properly generated as described in previous paragraph. For each query P in D_Q we compute the set $M_{P,t}$ containing all embeddings from P to \mathcal{F} . Then for each embedding e in $M_{P,t}$ we create an array K_e , denoted as *embedding-array*, as

```

- Map: <XML fragment  $\mathcal{F}$ , XPath query  $Q$  >
DQ = getDecomposition(Q); //Return the decomposition of  $Q$ 
NB = getBranchingNodes(Q); //Return the branching nodes of  $Q$ .  $|N_B|$  is the size of  $N_B$ 
nDB = get1stbranchingnode(Q); //Return the first branching node of  $Q$ .
For each query  $P$  in  $D_Q$  do
  For each embedding  $e$  from  $P$  to  $\mathcal{F}$  do
    Ke[j] = *; //Initialize the array  $K_e$  of the images of the branching nodes,
    of size  $|N_B|$  (i.e.,  $j = 0, \dots, |N_B| - 1$ ).
    For each branching node  $n$  of  $P$  appearing in the position  $\ell$  of  $N_B$  do
      Ke[ $\ell$ ] =  $e(n)$ 
      If  $P$  is the selection path of  $Q$  then
        output  $\leftarrow \langle e(n_{DB}), [K_e, P, e(out(Q))] \rangle$ 
      else
        output  $\leftarrow \langle e(n_{DB}), [K_e, P, true] \rangle$ 
- Reduce: < Key  $K$ , Collection  $Values$  >
DQ = getDecomposition(Q);
B = getBuckets(Values)
For each  $T$  in  $B$ 
  For each  $[c_1, c_2, c_3]$  in  $T$ 
    If  $c_2$  is either the selection path of  $Q$  or the query  $Q$ 
      output  $\leftarrow \langle K, c_3 \rangle$ 

```

Fig. 2. HoX-MaRe Algorithm

follows. For each branching node n of Q which is included in P we put the node $e(n)$ to the position $h(n)$ of K_e , while we put the symbol “*” to each position of K_e which is not mapped by a node of P .

To define the key of each pair we distinguish the first branching node n_B of Q , traversing the selection path of Q from the root to the output. It is easy to verify that this node is included in each query in D_Q . Finally, for each embedding e , the map function outputs a key-value pair, where the key is $e(n_B)$, and the value is a triple of the form $[K_e, true, P]$, when the query P is not the selection path of Q or a triple of the form $[K_e, out(P), P]$, otherwise. Note here that if the input query does not have any branching node the key of each pair is given by null values; consequently all pairs are routed in a single reducer.

Example 2. Consider the query Q , the XML tree t , the horizontal fragmentation \mathcal{T} of t and the decomposition D_Q of Q illustrated in Fig. 1. Notice that Q has 2 branching nodes; the n_1 and n_2 . Running the map function of the HoX-MaRe algorithm over \mathcal{T} , two map tasks are performed; one on each fragment. Applying the map function on \mathcal{F}_1 we compute the embedding e_1 from P_1 to \mathcal{F}_1 , where $e_1(n_0) = m_{12}$, $e_1(n_1) = m_7$, $e_1(n_2) = m_2$ and $e_1(n_4) = m_1$, as well as its embedding-array $K_{e_1} = [m_7, m_2]$. The embedding-array $K_{e_2} = [m_7, m_4]$ of the embedding e_2 from P_2 to \mathcal{F}_1 is computed similarly in the same task, while the second task computes the embedding-array $K_{e_4} = [m_7, *]$ of e_4 from P_4 to \mathcal{F}_2 . For K_{e_1} and K_{e_2} the first task outputs the key-values pairs $\langle m_7, [K_{e_1}, true, P_1] \rangle$ and $\langle m_7, [K_{e_2}, true, P_2] \rangle$, while the second task outputs the pair $\langle m_7, [K_{e_4}, m_9, P_4] \rangle$ for e_4 . We follow the same procedure for each embedding computed in each task.

Reduce Function: The reduce function performs over the input key-value pairs as follows. A function *getBuckets* is initially applied over the input values. This function groups properly the triples included in the value of the input pair, based on the concept of *unification* of the embedding-arrays. We say that two embedding-arrays K_1, K_2 are *unifiable* if there is not any integer i such that $K_1[i] \neq K_2[i]$ and $K_1[i], K_2[i] \neq *$. In particular, the function *getBuckets* returns every set B (denoted *bucket*) containing tuples such that for each query P in the decomposition of Q there is a tuple in B which describes an embedding from P and for every two tuples in B their embedding-arrays are unifiable. Then, for each bucket B , the reducer locates the tuple obtained by the selection path and outputs the output of the selection path.

Example 3. Continuing the Example 2 and considering that all the key-value pairs have been emitted from the mappers, it is easy to verify that there is a reduce task which receives all the pairs having m_7 as a key. The reduce task initially calls the function *getBuckets*. Notice that there is not any bucket returned by *getBuckets* which contains both the values $[K_{e_1}, true, P_1]$ and $[K_{e_2}, true, P_2]$, since K_{e_1} and K_{e_2} are not unifiable (notice that $K_{e_1}[1] \neq K_{e_1}[2]$). However, the values $[K_{e_1}, true, P_1]$ and $[K_{e_4}, m_9, P_4]$ will be contained in a returned bucket. For this bucket, the reduce function will output the image of the output of P_4 , which is given by the node m_9 .

3.2 Preliminary Experimental Results

In this section, we present a set of preliminary experiments performed on a Hadoop cluster of 14 nodes of the following characteristics: Pentium(R) Dual-Core CPU E5700 @ 3.00GHz, 4GB RAM and 30GB available disk space. We run the HoX-MaRe algorithm (see Fig. 2) over an XML document of 1.5GB given by the XML generator of the XMark project[1]. We assigned to each node of the XML document a new attribute indicating a unique ID value and posed a set of XPath queries over several horizontal fragmentations, in terms of maximum fragment size, as well as we run the queries using 4, 9 and 14 nodes. Our results are summarized in the Fig. 3. where the table includes the average of the evaluation time, in min, of the queries, for each fragment-size and each number of cluster nodes. Note that, the time values depicted in this table correspond to the total time for evaluating queries. From the results of the table we conclude that the more nodes we use the faster we get the result of a query; which shows the load balancing feature of the algorithm. Furthermore, we can easily notice that the evaluation time is reduced when we use small XML fragments. This can be explained by the fact that our implementation has been built using a DOM package which requires loading of the whole XML document, in each mapper, into memory. In order to compare our approach with the evaluation of XPath queries in a single computer we tried to run queries using the DOM package. However this was not possible for XML files greater than 100MB.

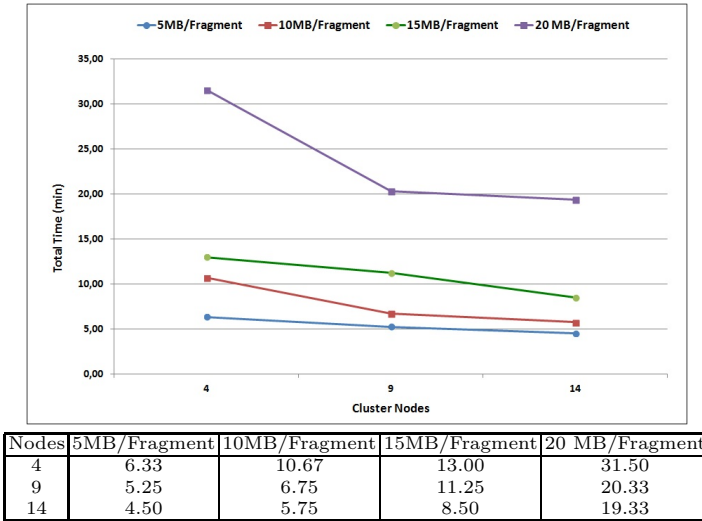


Fig. 3. Preliminary Experimental Results

4 Conclusions and Related Work

In this paper we presented an algorithm for distributed XPath query evaluation based on MapReduce. Our preliminary experiments shows that the algorithm is scales well to large XML datasets. As future work, we plan to investigate heuristics in order to improve further the performance of our algorithm, and make use of hash function in order to improve load balancing of the algorithm.

References

1. XMark: An XML Benchmark Project, <http://www.xml-benchmark.org>
2. Choi, H., Lee, K.-H., Kim, S.-H., Lee, Y.-J., Moon, B.: HadoopXML: a suite for parallel processing of massive XML data with multiple twig pattern queries. In: CIKM, pp. 2737–2739 (2012)
3. Cong, G., Fan, W., Kementsietsidis, A., Li, J., Liu, X.: Partial evaluation for distributed XPath query processing and beyond. *ACM Trans. Database Syst.* 37(4), 32 (2012)
4. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. *Commun. ACM* 51(1), 107–113 (2008)
5. Fegaras, L., Li, C., Gupta, U., Philip, J.: XML query optimization in Map-Reduce. In: *WebDB* (2011)
6. Garcia-Molina, H., Ullman, J.D., Widom, J.: *Database Systems: The Complete Book*. Prentice Hall Press, Upper Saddle River (2008)

7. Suci, D.: Distributed query evaluation on semistructured data. *ACM Transactions on Database Systems* 27, 2002 (1997)
8. Tatarinov, I., Viglas, S., Beyer, K.S., Shanmugasundaram, J., Shekita, E.J., Zhang, C.: Storing and querying ordered XML using a relational database system. In: *SIGMOD Conference*, pp. 204–215 (2002)
9. Zinn, D., Khler, S., Bowers, S., Ludscher, B.: Parallelizing XML processing pipelines via MapReduce. Technical report (2009)

A Query Language for Workflow Instance Data

Philipp Baumgärtel*, Johannes Tenschert, and Richard Lenz

Institute of Computer Science 6,
University of Erlangen-Nuremberg
{philipp.baumgaertel,johannes.tenschert,richard.lenz}@fau.de

Abstract. In our simulation project ProHTA (Prospective Health Technology Assessment), we want to estimate the outcome of new medical innovations. To this end, we employ agent-based simulations that require workflow definitions with associated data about workflow instances. For example, to optimize the clinical pathways of patients with stroke we need the time and associated costs of each step in the clinical pathway. We adapt an existing conceptual model to store workflow definitions and instance data in RDF. This paper presents a query language to aggregate and query workflow instance data. That way, we support domain experts in analyzing simulation input and output. We present a heuristic algorithm for efficient query processing. Finally, we evaluate the performance of our query processing algorithm and compare it to SPARQL.

1 Introduction

ProHTA (Prospective Health Technology Assessment) is a simulation project aimed at estimating the potential of innovative healthcare technologies at a very early stage. To this end, new types of hybrid and modular simulation systems are employed to simulate the effects of new healthcare technologies [5]. For example, one of our simulations concerns mobile stroke units [5]. For stroke, the time between onset and treatment is crucial for the treatment process. Mobile stroke units enable diagnosis and treatment of stroke patients on site, therefore reducing the time between onset and treatment. Hence, in our simulation models, we pay great attention to the diagnosis and treatment workflows of stroke patients and the time of individual steps in these workflows.

Besides the problem of simulation modeling, simulation input data management is an important concern [11]. Because medical and statistical simulation data in our project stems from several heterogeneous sources, a generic and flexible conceptual model is required. We developed a multidimensional conceptual model using RDF (Resource Description Framework) to cope with the heterogeneity [2].

In our simulation project, we are already using workflow definitions in form of activity diagrams [9] as a first step towards simulation models. Therefore, it is natural to organize our simulation input data according to these workflows. To

* On behalf of the ProHTA Research Group.

this end, activity diagrams need to be stored in the data management system. Then, simulation input and output data can be stored and linked to the activity diagrams.

Our simulation practitioners and domain experts want to query and analyze data. However, it is hard for scientists to write non-trivial SQL or SPARQL queries [10]. Therefore, we propose using a domain specific query language. In our stroke example, the simulation estimates the time between onset and treatment gained by implementing mobile stroke units. Then, the domain experts could use a domain specific query language to compare the simulation outcomes of different settings.

Together with our simulation experts, we identified several requirements for such a domain specific query language:

1. *Query aggregated data for a specific part of a workflow*
2. *Query data for individual steps of a workflow*
3. *Query aggregated data for the most probable paths through a workflow*

In this paper, we present a conceptual model to organize data according to workflow definitions. Additionally, we develop a domain specific query language to query and analyze the data.

2 Conceptual Model

Dumas and Hofstede [7] evaluated UML activity diagrams as a specification language for workflows. Despite some imprecise semantics, they satisfy all of our requirements. As we are already using RDF to store multidimensional data [2], we decided to store UML activity diagrams using RDF. Dolog’s OWL ontology allows for storing UML state machines in RDF [6]. As UML activity diagrams can be mapped to UML state machines, we use a simplified version of Dolog’s ontology.

The basic elements of activity diagrams are depicted in Fig. 1. There are states and transitions, for example “A” and “C” are simple states. Transitions are depicted as arrows. Additionally, there are initial and final states. Composite states can be used to construct hierarchical structures and may contain parallel regions. Branches like “B” can be used to indicate alternatives.

We decided to omit forks and joins because the well-formedness of diagrams containing forks and joins is non-trivial [7]. However, parallel execution can still be achieved without losing expressiveness by using parallel regions in composite states. The execution of a composite state is complete when it reaches all final states in the parallel regions of the composite state. A transition between a state inside a composite state and a state on the outside interrupts the execution of the composite state. For example, the transition between “B” and “C” interrupts the execution of the composite state in Fig. 1.

We extended Dolog’s ontology with optional probabilities for outgoing transitions of branches. Additionally, we can store the time and different types of costs of states and transitions. These costs consist of a name and a numerical value. In addition to the workflow definition with aggregated data, we store data

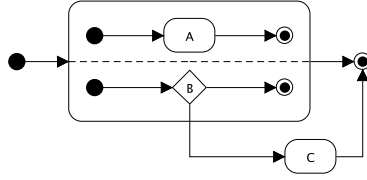


Fig. 1. UML activity diagram

about workflow instances. For example, we store data about the treatment of a patient. This fine-grained data can be used for query evaluation instead of the preaggregated data stored alongside the workflow definition.

3 Workflow Query Language

In this section, we present the main features of the WQL (Workflow Query Language). To this end, we introduce the ESTIMATE query type that can be used to aggregate time and costs in workflow definitions. We provide a formal definition of workflow data aggregation semantics online¹. The scheme of an ESTIMATE query is shown in listing 1.1. Paths with a denoted start and end are examined in order to aggregate time and costs.

```

[ CONTEXT <URI> ]
ESTIMATE time, costs, probability, state, path
OF <workflow>
[ FROM <start> ] [ TO <end> ]
[ USING INSTANCE named instance ]
[ USING INSTANCES named instances for average times ]
[ USING ALL INSTANCES ]
[ WITH { variables, times, decisions } ]
[ GROUP BY state, path ]
[ ORDER BY time, costs, probability ASC/DESC ]
  
```

Listing 1.1. Scheme of an ESTIMATE query

To prevent the user from having to write the same prefix of URIs multiple times, the CONTEXT statement allows an abbreviated form similar to 'PREFIX' in SPARQL. ESTIMATE queries allow multiple column definitions in the ESTIMATE clause, e.g. time and different types of costs. Also, states, paths and the probability of paths can be queried if states or paths are part of the GROUP BY clause. Then, GROUP BY works like its SQL counterpart.

The optional FROM and TO clauses specify the beginning and end of the considered paths. Initial and final states of the examined workflow are the default values for FROM and TO. States can be identified either by URI or by unique

¹ http://www6.cs.fau.de/people/philipp/wql_semantics.pdf

names. The USING (ALL) INSTANCE(S) and WITH clauses specify workflow instances as described in Sect. 2 for the aggregation of time and costs. The ORDER BY statement triggers sorting of results with the desired sort order.

4 Query Processing

As the expressiveness of SPARQL is not sufficient, we cannot translate ESTIMATE queries to SPARQL. Therefore, we use SPARQL only to load data and activity diagrams and provide a custom query processing algorithm. In this section, we present the path-finding algorithm to process ESTIMATE queries. Since loops and decisions can produce an infinite number of paths, finding all of them is impossible. Hence we present a heuristic approach for finding the most likely paths. First, we present the basic algorithm that is not able to handle parallel sections. After that, we describe the extensions to support parallelism.

Since our path-finding algorithm is a heuristic, three parameters are provided to limit processing: the minimal probability of a path p_{\min} , the maximal number of results r_{\max} , and the maximal number of states in a path n_{\max} . Algorithm 1 shows a simplified version of our algorithm. In the following, we call the transitions of the activity diagram edges. The function `suitableEdges(state)` returns all outgoing transitions of a state excluding transitions to states with no path to a final state and transitions excluded by conditions. Therefore, we need to mark each state that reaches the end in advance.

Algorithm 1. Path-finding heuristic

```
List result, PriorityQueue pq
setCapacity(pq,  $r_{\max}$ )
enqueue(pq, start, priority = 1)
while ¬empty(pq):
    path = pop(pq)
    edges = suitableEdges(last(path))
     $\forall edge \in edges$ :
        if length(path + edge) >  $n_{\max} \vee$  probability(path) · probability(edge) <  $p_{\min}$ :
            continue
        if reachEnd(path + edge):
            append(result, path + edge)
            setCapacity(pq,  $r_{\max}$  - length(result))
        else:
            enqueue(pq, path + edge, priority = probability(path) · probability(edge))
```

The priority of a path in the priority queue is simply the probability of the path. Therefore, we try all suitable outgoing edges of the last state in the path with the highest probability. If none of our aforementioned limits is exceeded, we create new paths for each outgoing edge of the last state in that path. We append these new paths to the priority queue if they do not reach the end. Otherwise, we append them to the result list.

Our algorithm is used recursively for each parallel compound state to support nested parallel compound states. Resulting paths are no longer sequences as we have to store the states in each parallel region of the compound state. We store for each path whether it interrupts parallel execution or not. All non-interrupting paths are put into the priority queue.

For each interrupting path the paths of all other parallel regions have to be aborted at a certain point. To determine this point, we use the time of the interrupting path and search for all paths in the parallel regions with a shorter time span.

5 Evaluation

The evaluation of our heuristic is divided into two parts. First, we present an acyclic worst-case scenario and evaluate it against SPARQL property paths². Afterwards, we evaluate a cyclic activity diagram and assess the precision of results. Our prototype is written in Python. SPARQL queries are processed by Fuseki 0.2.1.

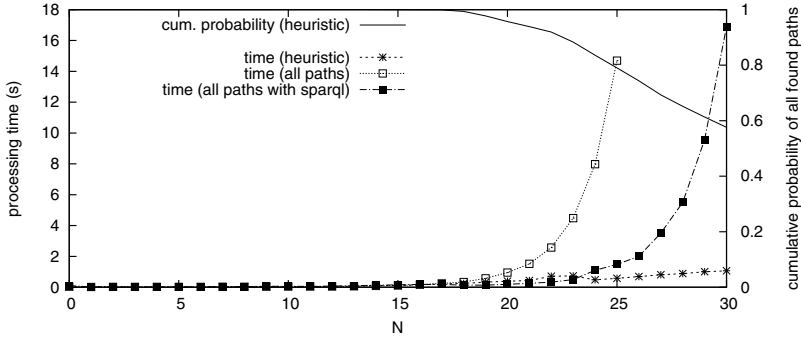
As SPARQL supports property paths to query paths in RDF graphs, we want to compare them to our path finding heuristic. These property paths are like regular expressions for RDF properties and can be used to query paths of arbitrary length in an RDF graph. SPARQL only finds matching endpoints to a property path and does not search for all paths between these two endpoints. Therefore, property paths aren't suitable for finding all paths in activity diagrams. However, for evaluating acyclic activity diagrams without parallelism with SPARQL we can simulate the search for paths. To this end, we enumerate all paths between two endpoints and store each path with separate synthetic endpoints in RDF. Then, we can write SPARQL queries using property paths that find and return the endpoints of all paths. Hence, the complexity of processing these SPARQL queries can be compared to our path finding algorithm. However, even with this extension, SPARQL property paths would not be applicable to cyclic diagrams or parallel regions.

Fig. 2(b) shows an example for a “Fibonacci activity diagram”. In these synthetic diagrams with N states, each state is connected to its two successors. We evaluated ESTIMATE queries asking for all paths that start at the initial state and end at the final state of the activity diagrams. The number of paths for each diagram with N states is the corresponding Fibonacci number, so an exponentially growing quantity of paths is generated. We define that transitions to direct successors (e.g. B to C) of a state have a probability of 90%.

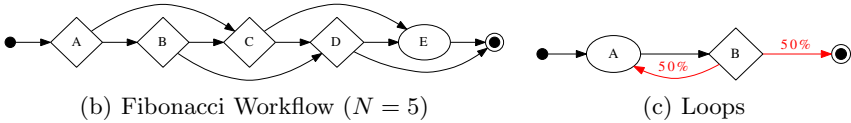
Our heuristic tries to find $r_{\max} = 2000$ most probable paths. Fig. 2(a) shows the cumulative probability of all found paths and the time to process each query. As expected, at some point the cumulative probability of the found paths decreases.

Fig. 2(a) shows that processing all possible paths is only appropriate to a certain extent. The time to find all paths using our SPARQL workaround or our

² <http://www.w3.org/TR/sparql11-property-paths/>



(a) Comparison of processing times



(b) Fibonacci Workflow ($N = 5$)

(c) Loops

Fig. 2. Processing times and evaluated workflows

path finding algorithm (searching for all paths) grows exponentially. SPARQL-only processing with property paths is faster than an exhaustive search for all paths with our algorithm as our implementation in python is not very fast. Despite of this, a heuristic search with our algorithm is much faster as it tries to find only the most probable paths. By defining the limits of the heuristic, the user is able to balance processing time and path coverage, which is depicted as cumulative probability of found paths.

Not all paths are equally important for results. Therefore, our heuristic tries to find the r_{\max} most probable paths. For cycles, longer paths usually have less probability. Hence, in cyclic diagrams a few paths cover the most probable scenarios.

Fig. 2(c) shows an example of a loop with $cost(A) = 10$ and $cost(B) = 0$. Processing is limited by r_{\max} . Since this is a simple example, the precise average cost of all paths can be determined:

$$cost = \sum_{i=1}^{\infty} \frac{1}{2^i} \cdot 10i = 10 \sum_{i=1}^{\infty} i \cdot 2^{-i} = 10 \cdot 2 = 20 \tag{1}$$

By accumulating the weighted cost of the 10 most important paths, a relative error of $5.86 \cdot 10^{-3}$ remains. At 25 paths, the relative error is $4.02 \cdot 10^{-7}$ and therefore negligible. This is because for cycles longer paths usually have less probability. The query for 25 paths took 0.042s. Hence, our heuristic is well-suited for cyclic diagrams.

6 Related Work

Awad [1] reviews existing query languages for business processes and classifies them according to three categories:

1. Querying business process definitions
2. Querying running instances of business processes
3. Querying execution history (logs) of completed business processes

The query languages in the first category are concerned with querying the structure of business processes. The approaches in the second category monitor running business processes. The third category is known as workflow or process mining. As our approach is concerned with querying the definition of a workflow, we consider the WQL to be in the first category. However, we also need to query data associated with the workflow definition. Therefore, we review some existing approaches in that category in addition to the literature already listed by Awad.

Awad [1] proposes a visual query language to search repositories of business processes for certain patterns. Deutch and Milo [4] provide a comprehensive formalism to study process modeling and querying. They use this formalism to evaluate the BPQL (Business Process Query Language) [3]. Francescomarino and Tonella [8] define the semantics of a visual query language for business processes by translating queries to SPARQL. They deal with the problem of querying paths between two elements. However, they assume each pair of elements that is connected by a path to be directly connected by an RDF property `p:isConnectedTo`. Hence, their solution is much simpler than our path finding algorithm. The aforementioned approaches including the approaches listed by Awad do not consider the data perspective and do not allow for aggregation of data. Therefore, our approach provides some unique contributions in this regard.

7 Conclusions and Future Work

In this paper, we presented a language to query aggregated data. To this end, we adapted an existing conceptual model to store workflow definitions as activity diagrams in combination with workflow instance data in RDF. We developed a heuristic query processing algorithm and evaluated it in comparison with pure SPARQL. Our evaluation shows that our heuristic algorithm is well suited for complex workflow definitions and is able to cope with cyclic graphs. The query language enables our domain experts to analyze simulation input and output data. Additionally, we can use this query language to load input data into our simulation models. Therefore, our query language renders both the input data management process and the evaluation of simulation output data more efficient.

In future work, we are planning to implement a semi-automatic transformation from activity diagrams to agent-based simulation models. Therefore, the conceptual model, aggregation formalism and query language will become more integrated with our simulation tools. We need to extend our conceptual model to support probability densities instead of fixed times and costs. Moreover, we

will investigate how to combine this conceptual model with our existing multidimensional conceptual model [2] to support data that depends on various factors, e.g. the age of a patient.

Acknowledgements. This project is supported by the German Federal Ministry of Education and Research (BMBF), project grant No. 13EX1013B.

References

1. Awad, A.: BPMN-Q: A Language to Query Business Processes, vol. 119, pp. 115–128. Citeseer (2007)
2. Baumgärtel, P., Lenz, R.: Towards data and data quality management for large scale healthcare simulations. In: Conchon, E., Correia, C., Fred, A., Gamboa, H. (eds.) Proceedings of the International Conference on Health Informatics, pp. 275–280. SciTePress - Science and Technology Publications (2012) ISBN: 978-989-8425-88-1
3. Beeri, C., Eyal, A., Kamenkovich, S., Milo, T.: Querying business processes. In: Proceedings of the VLDB 2006 (2006)
4. Deutch, D., Milo, T.: A structural/temporal query language for business processes. *Journal of Computer and System Sciences* 78(2), 583–609 (2012)
5. Djanatliev, A., Kolominsky-Rabas, P., Hofmann, B.M., Aisenbrey, A., German, R.: Hybrid simulation approach for prospective assessment of mobile stroke units. In: SIMULTECH 2012 - Proceedings of the 2nd International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 357–366 (2012)
6. Dolog, P.: Model-driven navigation design for semantic web applications with the uml-guide. In: Matera, M., Comai, S. (eds.) *Engineering Advanced Web Applications*. Rinton Press (2004)
7. Dumas, M., ter Hofstede, A.H.M.: UML activity diagrams as a workflow specification language. In: Gogolla, M., Kobryn, C. (eds.) *UML 2001*. LNCS, vol. 2185, pp. 76–90. Springer, Heidelberg (2001)
8. Di Francescomarino, C., Tonella, P.: Crosscutting concern documentation by visual query of business processes. In: Ardagna, D., Mecella, M., Yang, J. (eds.) *Business Process Management Workshops*. LNBIP, vol. 17, pp. 18–31. Springer, Heidelberg (2009)
9. Gantner-Bär, M., Djanatliev, A., Prokosch, H.U., Sedlmayr, M.: Conceptual modeling for prospective health technology assessment. In: Proceedings of the XXIV Conference of the European Federation for Medical Informatics (2012)
10. Howe, B., Cole, G., Souroush, E., Koutris, P., Key, A., Khoussainova, N., Battle, L.: Database-as-a-service for long-tail science. In: Bayard Cushing, J., French, J., Bowers, S. (eds.) *SSDBM 2011*. LNCS, vol. 6809, pp. 480–489. Springer, Heidelberg (2011)
11. Skoogh, A., Johansson, B.: A methodology for input data management in discrete event simulation projects. In: Proceedings of the 40th Conference on Winter Simulation, WSC 2008, pp. 1727–1735 (2008)

When Too Similar Is Bad: A Practical Example of the Solar Dynamics Observatory Content-Based Image-Retrieval System

Juan M. Banda, Michael A. Schuh, Tim Wylie,
Patrick McInerney, and Rafal A. Angryk

Montana State University, Bozeman, MT 59717 USA
{juan.banda,michael.schuh,timothy.wylie,
patrik.mcinerney,angryk}@cs.montana.edu

Abstract. The measuring of interest and relevance have always been some of the main concerns when analyzing the results of a Content-Based Image-Retrieval (CBIR) system. In this work, we present a unique problem that the Solar Dynamics Observatory (SDO) CBIR system encounters: too many highly similar images. Producing over 70,000 images of the Sun per day, the problem of finding similar images is transformed into the problem of finding similar solar events based on image similarity. However, the most similar images of our dataset are temporal neighbors capturing the same event instance. Therefore a traditional CBIR system will return highly repetitive images rather than similar but distinct events. In this work we outline the problem in detail, present several approaches tested in order to solve this important image data mining and information retrieval issue.

1 Background and Motivation

Content-based Image-Retrieval (CBIR) systems are imperative in many research areas and industries where the amount of information to sort, search, and retrieve is greater than what is humanly feasible. CBIR systems are currently used across many diverse fields such as medical vision, video surveillance, law enforcement, facial recognition, tracking, and more [10,12,14,6].

How the CBIR systems are used and how they work also vary depending on the needs of the application. Some systems are designed to find identical visual characteristics, while others focus on finding structural similarity. Defining what is of interest is an important aspect of a CBIR system. This measure is application dependent and guides what techniques can be used and how the data is processed. CBIR systems are well-known with methods having been developed which index and define interest based on color features [11], shapes of certain objects [8], textures [6], etc.

The Solar Dynamic Observatory (SDO) mission was launched in 2011 and captures 70,000 images a day over 10 wavebands providing an unprecedented 1.5 TB a day of information about the Sun. The exponential growth of cross comparison between all images makes most standard methods of comparison

infeasible. Therefore we must address a more important issue that is often overlooked in most systems since none of them, to our knowledge, have to deal with the same volume of highly similar data.

When studying the Sun, solar physicists primarily study solar phenomena that we will refer to as solar events. The type of events that are of interest can vary greatly in size, duration, location, and in the colocation possibilities with other events. Thus, we have many different types of tasks to perform in this CBIR system that no other traditional system performs. Our initial attempts to tackle this ambitious problem consist of several methods:

- We will focus on traditional nearest-neighbor retrieval to fully exemplify our problem and decide on a few starting points for more potential experiments.
- We experiment on our CBIR system by using the labels to attempt to solve the temporal cadence issue. These labels all come from central location, Heliophysics Event Knowledgebase (HEK), where all the Feature Finding Team (FFT) modules report to.
- When searching for similar events, the most similar images are from the proceeding and succeeding timesteps which also contain the same event we are querying on. This makes finding patterns or similar singular events difficult. We can intuitively reduce this by increasing the cadence of the system. Unfortunately, this results in excluding many important short duration events such as solar flares as we will show in the experiments section.

In this paper we introduce the problem of finding similar events in a temporal dataset. We discuss several ways to approach the problem and show how effective they result on our test dataset, and we also lay out some possible future directions to improve CBIR systems that face these similar issues.

2 Experimental Setup

2.1 Image Parameters

As we have presented in our previous works, we use some of the more popular image parameters that are used in fields such as medical imaging, natural scene images, and traffic imaging [12,14,6]. We use a grid-based image segmentation with 4,096 cells per image shown to be the most effective on our solar images [6]. The ten image parameters that we have defined to be the most useful [3] are: Entropy, Mean, Standard Deviation, 3rd Moment (skewness), 4th Moment (kurtosis), Uniformity, Relative Smoothness (RS), Fractal Dimension, Tamura Directionality, and Tamura Contrast, more details on them can be found on [4].

2.2 Filtering Mask

To identify regions of interest for active solar events, we employ a simple intensity-based region growing technique [7]. Pixels of intensity greater than the 99.5 percentile for the image are selected as the 'seeds' of the regions [1]. The regions are then grown by iteratively adding any pixels of intensity above the 80th percentile

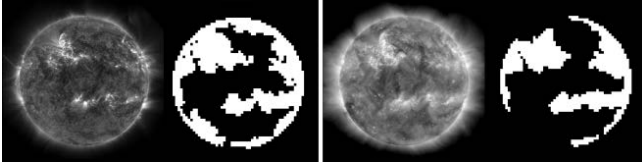


Fig. 1. Regions produced by the mask for wavelengths 171(left) and 193(right)

of the image that are 8-way adjacent to the current region. Terminating once no more pixels can be added. Next we apply a radial filter to the image, eliminating all pixels that are not within a fixed distance (the Sun’s radius) of the image center. Finally, we build the set of grid cells that contain one or more pixels of the remaining regions, resulting in the white images to the right in Figure 1.

2.3 Similarity Measures

In order to determine if we can have more useful and interesting similarities between images/events, we utilize ten different distance metrics that will showcase different properties between our images/events. These metrics are addressed in detail in [4], and include: Euclidean, Std. Euclidean, City Block, Chebyshev, Cosine, Correlation, Spearman, and Fractional Minkowski with $p = 0.5, 0.75, 0.90$.

2.4 The SDO Dataset

To create an SDO dataset we had to overcome one problem: finding annotated event data. Since asking experts to manually annotate 4k by 4k resolution images is unrealistic, we had to wait for several of the modules of the Feature Finding Team (FFT) of the SDO mission [13] to be fully running and reporting their results on their respective solar events they were designed to detect. This dataset consists of images from four different wavebands over a three-day period (from January 20, 2012 to January 23, 2012, subset of the one presented in [5]) where there was a representative amount of solar activity resulting in multiple occurrences for each type of event. We experiment with four different AIA wavebands (94, 131, 171, 193), and four types of labeled events: Active Region (AR), Coronal Hole (CH), Flare (FL), and Sigmoid (SG), with 292, 71, 161, and 95 event labels respectively. Each one of these events are reported by a module of the FFT that has been independently developed by a specialized team of solar physicists and computer scientists using image processing, statistical analysis, and data mining techniques [13].

The original version of this dataset can be found here [2]. In order to present our unique interestingness and relevance problem we will use four different versions of this dataset that are outlined in Table 1.

Table 1. Original, DS2 and DS3 will be split by wavelength for our experiments

Label	Description	Images
Original	Four wavelengths, time cadence of 6 minutes	3,394
DS1	One image per labeled event	619
DS2	Original dataset with mask from section 2.2	3,394
DS3	One image per labeled event and mask from Section 2.2	619

2.5 Experiment Descriptions

Experiment 1. Using dataset Original, we calculate each images nearest neighbors for each different wavelength. We also performed the same calculation using all ten different distance metrics from section 2.3. Finally, we will plot all the images and their nearest neighbors sorted by their time stamp from left to right. With this experiment we show how the temporal aspect of our data is affecting the similarity problem, by returning as the closest neighbors all the closest temporal images.

Experiment 2. Using dataset DS1, we generate a similarity graph for each different event with all its possible nearest-neighbors from the same event type, generating a different graph for each distance metric. We then plot the events and the distances to others sorted by event time stamp from left to right. With this experiment we expect to see if the similarity effect of the temporal repetition of the images is reduced when we group sets of images in events, based on their time ranges. Ideally, we would see similarity plots that contain the most-similar events from the time range and not an ordered list of events by time stamp.

Experiment 3. Using dataset Original, we calculate each image nearest neighbors for each wavelength, but we add a time step component (sampling cadence). We also perform the same calculation using all ten different distance metrics. Finally, we plot all the images and their nearest neighbors sorted by their time-stamp from left to right. By increasing the time cadence of sampled images from our dataset, we expect to lower the image repetition and have more interesting nearest neighbors than before. While seen as the most intuitive solution, this approach will introduce other problems.

Experiment 4. Using the same set-up as Experiments 1, 2, and 3, but with datasets DS2, DS3, and DS2, respectively, we again plot the images and their nearest neighbors sorted by time-stamp from left to right. In order to reduce the storage expense and remove uninteresting parts of the solar image in an automated way, we apply the mask described in Section 2.2 to get any performance gains from considerably reduced datasets that now only contains regions of interest.

3 Results and Analysis

This section contains the most interesting results for the previously outlined experiments. If you are interested in seeing all of the resulting plots, or in replicating the experiments, please visit the supplemental website [2].

3.1 Experiment 1

Testing every image in the dataset separated by wavelength will allow us to observe the temporal similarity relation between the images nearest neighbors. Our similarity (a.k.a nearest neighbor) matrix is plotted in a symmetrical way and all our distance values are scaled from 0 to 1, with 0 being closer and 1 being the farthest away. The colors of our plot range between dark blue to dark red and they represent the same 0 to 1 scale, respectively.

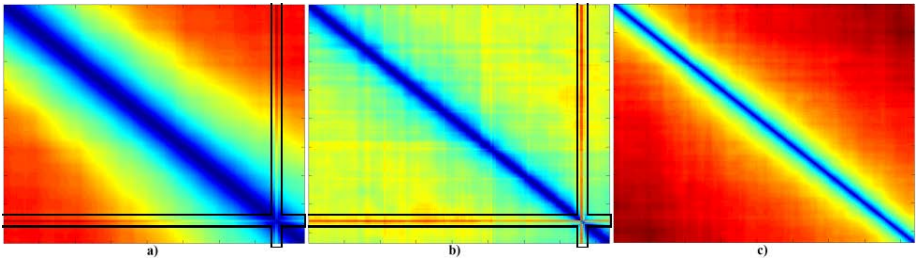


Fig. 2. Nearest neighbor plots for 131 a) Sperman distance and param. mean, b) Correlation distance param. tamura contrast 10. 171 c) City block distance and param. standard deviation.

As we can see, the temporal similarity corresponds to the distance levels as they change from blue to red. The problem is that almost all behave in the same manner—it is blue when it is close temporally. From the range of the dark blue we can see the closest neighbors are the immediate temporal images. This behavior is consistent, except for the boxed region in Figure 2 a and b, which region/timeframe corresponds to a large solar flare. In this event the intensity of the solar values goes very high for a short period of time and drastically fluctuates between consecutive images and thus the red streaks inside the selected area. However, as the event ends, the behavior returns to normal again having the events be very similar in terms of distance with respect to time.

Keeping with the consistent behavior of shifting from blue to red as the time stamp increases, we show that almost any combination of image parameter and distance metric will be a victim of this similarity problem. The flare event outline is found on Figure 2 a, b, with only c (city block distance with param. standard deviation) eliminating its presence. This will quickly lead us to discarding the combination since we will lose an event we are trying to find.

3.2 Experiment 2

Taking one image per solar event, we can reduce our dataset from 3,394 to 619 images since each event has a duration window ranging from minutes to several hours depending on the FFT module and their event-specific reporting standards, more on this can be found in [13]. With such reduction, we expected to have the time repetition factor less apparent.

Figure 3 a) and b) show the problematic behavior with this approach. As expected, the diagonal is 0 or blue, and the spreading out pattern goes from blue to light green (around 0.29 according to the scale), which indicates that most active region events that are similar are consecutive (within their wavelength) in time which is not useful for researchers trying to find similar events at a different time. Note this plot features the events sorted by time-stamp, not similarity, and features two different wavelengths as seen from the two sections in a) and b).

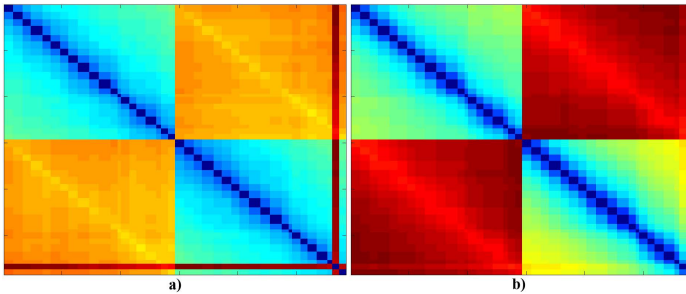


Fig. 3. Nearest neighbor plots for (by rows) AR a) Chebyshev distance and param. kurtosis, b) Euclidean distance param. standard deviation.

It is worth mentioning that there are two events without any real or temporal nearest neighbors indicating a completely different event occurrence. The flare outlined in Figure 2 is one example. From the labels provided by the FFT modules, we occasionally find inconsistencies like this. This furthers our emphasis of not relying only on labels for proper functionality of our CBIR system.

3.3 Experiment 3

The problem of temporally-dependent nearest neighbors is well showcased in Experiments 1 and 2 (and Figures 2 and 3). In our next experiment, we attempt to solve this problem by increasing the time cadence of sampled images, from the original 6 minutes to 18 to 60 minutes respectively. The results are more promising than for the similarity plots, but introduce one very critical issue that we will outline in the following figures.

Increasing the cadence allows some events to completely disappear, causing our system to miss some potentially relevant results. As we can see in the event

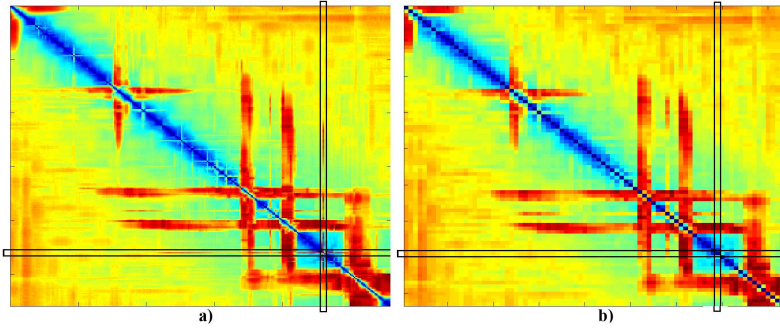


Fig. 4. Wavelength 94 nearest-neighbor plot for Chebyshev distance and entropy parameter with times step a) 18 minutes, and b) 60 minutes

found in Figure 4 a). When the time cadence increases this event fully disappears on the 60 minute cadence (b). While increasing the time cadence is a naïve and intuitive solution to reduce temporal repetition in nearest neighbors in a traditional CBIR system, this may not be ideal or useful for our Solar CBIR system, since it causes problems for short lived solar events that occur rapidly.

3.4 Experiment 4

While the masked versions of the Original dataset provided nearly identical results when used in Experiment 1 and 3, the most interesting and revealing results came for the masked version of Experiment 2.

In Figure 5 we have a clear example of how the mask allows our similarity results to change. The sigmoid events reported on the 131 wavelength are all similar on the DS1 dataset, but after the mask is applied for DS3, we can now differentiate them effectively. This makes a strong case that when grouping by events, there is a benefit to using an image mask to determine regions of interest. Another interesting behavior is that we are now able to see the event similarities

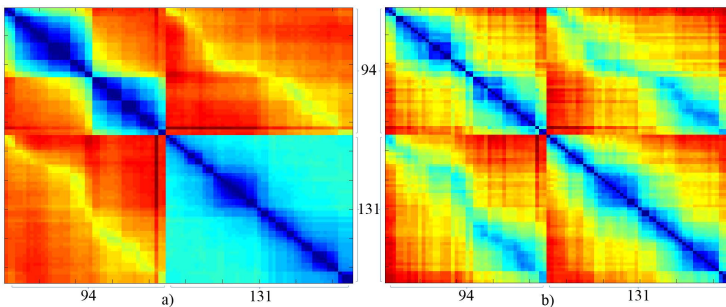


Fig. 5. SG event plot. DS1 dataset on the left, DS3 dataset on the right

across wavelengths— something that before was unlikely. This can be seen when comparing the upper right quadrants of a) and b) in Figure 5.

4 Conclusions

In this work we have outlined multiple intricacies of dealing with a dataset that has very similar images. Our problem also lies within the fact that our images are 4,096 by 4,096 pixels. When these two factors combine, we have determined that even when using different distance metrics, we will not be able to successfully analyze our image database without some kind of region of interest (ROI) approach that helps narrow down the query area and perform the image comparison at a lower level.

The results of Experiments 1, 2, and 3 indicate that the interestingness of our retrieval results cannot easily be solved by grouping images together based on event labels (Experiment 2) or by increasing the time cadence (Experiment 3). Since little literature exists that shows a CBIR system with this unique problem, it makes it an interesting data mining problem. There are few other known datasets that also have this problem, but we hypothesize that any video retrieval system would have a similar problem.

While the masked dataset DS2 did not provide any insightful results with Experiments 1 and 3, we did see an interesting development when used in conjunction with event based grouping (Figure 5 in Experiment 4). This leads us to believe that with the right combination of event type, wavelength, distance metric, and image parameter, we can still improve the differentiation of time-independent nearest neighbors.

We can conclude with the experiments performed and the analysis of the results, that with a hybrid approach combining time cadence reduction, interesting region mask, and the event grouping by waveband, distance metric, and parameter combination we would be able to see improvements in the returned nearest neighbors.

5 Future Work

We have started investigating practical and scalable solutions to region-based queries for our solar CBIR system. We expect the diversity of regions to diminish the problem of finding too similar results, but it will not be entirely eliminated. While we cannot use brute-force similarity-matching on dynamic regions for the full-scale system, it could provide us a benchmark of performance on a small sample dataset while working towards provably better solutions.

One possible direction for improvement is the incorporation of pre-defined region segmentation through clustering and classification on known data characteristics which could help reduce the search space of typical queries through pruning large quantities of unwanted regions. If a user is querying a ROI that resides in a bright region, we can eliminate all other regions while maintaining a high likelihood of returning similar region results with similar events.

More advanced solutions are still needed, and current directions of interest include using variations on recent visual bag-of-words approaches, or exploring hybrid indices that combine different data characteristics to achieve a singular comprehensive index. Thus, our existing full-image similarity-based indices must be extended to regions with spatial and temporal contexts.

References

1. Adams, R., Bischof, L.: Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.* 16(6), 641–647 (1994)
2. ADBIS. Adbis 2013 website (2013), <http://www.jmbanda.com/ADBIS2013>
3. Banda, J.M., Angryk, R.A.: On the effectiveness of fuzzy clustering as a data discretization technique for large-scale classification of solar images. In: *IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE*, pp. 2019–2024 (August 2009)
4. Banda, J.M., Angryk, R.A.: Usage of dissimilarity measures and multidimensional scaling for large scale solar data analysis. In: *Proc. of the 2010 Conf. on Intelligent Data Understanding, CIDU 2010*, pp. 189–203 (October 2010)
5. Schuh, M.A., Angryk, R.A., Pillai, K.G., Banda, J.M., Martens, P.: A Large-Scale Solar Image Dataset with Labeled Event Regions. In: *20th IEEE Int. Conf. on Image Processing, ICIP 2013* (to appear, 2013)
6. Banda, J.M., Angryk, R.A., Martens, P.C.: On the surprisingly accurate transfer of image parameters between medical and solar images. In: *18th IEEE Int. Conf. on Image Processing, ICIP 2011*, pp. 3669–3672 (September 2011)
7. Benkhalil, A., Zharkova, V., Zharkov, S., Ipson, S.: Active region detection and verification with the solar feature catalogue. *Solar Physics* 235, 87–106 (2006)
8. Gagaudakis, G., Rosin, P.L.: Incorporating shape into histograms for cbir. *Pattern Recognition* 35(1), 81–91 (2002)
9. Jing, F., Li, M., Zhang, L.: Learning in region-based image retrieval. In: Bakker, E.M., Lew, M., Huang, T.S., Sebe, N., Zhou, X.S. (eds.) *CIVR 2003*. LNCS, vol. 2728, pp. 199–204. Springer, Heidelberg (2003)
10. Kulkarni, S., Verma, B.: Fuzzy logic based texture queries for cbir. In: *Proc. of the 5th Int. Conf. on Computational Intelligence and Multimedia Applications, ICCIMA 2003*, pp. 223–238. IEEE Computer Society, Washington, DC (2003)
11. Lei, Z., Fuzong, L., Bo, Z.: A CBIR method based on color-spatial feature. In: *Proc. of the IEEE Region 10 Conf., TENCN 1999*, vol. 1, pp. 166–169 (1999)
12. Lin, H.-C., Chiu, C.-Y., Yang, S.-N.: Linstar texture: a fuzzy logic cbir system for textures. In: *Proc. of the 9th ACM Int. Conf. on Multimedia, MULTIMEDIA 2001*, pp. 499–501. ACM, New York (2001)
13. Martens, P.C.H., Attrill, G.D.R., Davey, A.R., Engell, A., et al.: Computer vision for the Solar Dynamics Observatory (SDO). *Solar Physics* 275, 79–113 (2012)
14. Thumfart, S., Heidl, W., Scharinger, J., Eitzinger, C.: A quantitative evaluation of texture feature robustness and interpolation behaviour. In: Jiang, X., Petkov, N. (eds.) *CAIP 2009*. LNCS, vol. 5702, pp. 1154–1161. Springer, Heidelberg (2009)

Viable Systems Model Based Information Flows

Marite Kirikova¹ and Mara Pudane²

¹ Department of Systems Theory and Design
Riga Technical University, Latvia

² Institute of Applied Computer Systems,
Riga Technical University, Latvia

marite.kirikova@cs.rtu.lv, mara.pudane@rtu.lv

Abstract. In information systems engineering it is important to ensure that all essential information flows are properly identified and supported. Usually the set of relevant information flows is detected using such knowledge acquisition techniques as interviews and document analysis. Since nowadays flexibility, adaptability, and agility are important features of enterprises for their operational strength in a highly turbulent environment, the research question arises whether there is a generic set of requirements that have to be met by information systems to obtain and sustain above mentioned enterprise features. To answer this question, Viable Systems Model (VSM) is experimentally used as a basis for identification of a set of information flows, which should be present in VSM complying enterprises. Specific constructs presented in the enterprise architecture description language are proposed for consistent integration of detected flows into enterprise information systems.

Keywords: Viable Systems Model (VSM), information system, data flow, information flow, knowledge flow, ArchiMate.

1 Introduction

In the 21th century information systems have to support high variety of enterprise activities, since the enterprises must be inventive, humanistic, cognitive, community-oriented, liquid, agile, sensing, global, and sustainable [1]. To possess these features, enterprises need appropriate models of functioning, which are properly supported by advanced information technologies (IT). A Viable Systems Model (VSM) has been reported as one of the alternatives for highly competitive enterprise models [2], [3], [4]. The VSM is rooted in ideas of cybernetics and comprises five mutually related systems at one or several fractal levels [4].

Commonly the VSM is investigated from the economic, managerial, and organizational perspectives. In this paper, the research is done from the information systems engineering perspective. VSM is, in essence, a fractal system [2] and to some extent the paper is a continuation of the work on fractal information systems [5], [6], [7], [8]. Important drivers for this research were (1) work of J.P. Rios [4], which provides a comprehensive model of communication channels of VSM; and (2) the enterprise

architecture modeling language ArchiMate [9] - an expressive and flexible tool enabling graphical representation of many features of VSM. The research benefited also from the work on information logistics [10] that emphasizes the role of functional perspective in analysis of information flows.

The aim of the paper is to identify generic information flows, which are mandatory enterprise information flows from the point of view of VSM. For this purpose the functions and information channels of VSM are analyzed and enterprise architecture construct based information flow analysis approach is proposed. This approach helps to move towards well supported information circulation in enterprises, which use VSM as their basic model of functioning.

The paper is structured as follows. Section 2 reflects the related work that forms the basis of the research. Section 3 discusses flows in VSM and presents how the template for generic flows can be created. Section 4 proposes generic and specific constructs represented in enterprise architecture description language. These constructs can be used for data, information, and knowledge flow analysis. The specific approach for VSM based flow analysis is also proposed in Section 4. Section 5 provides brief conclusions and directions for further research.

2 Related Work

Viability is a capacity of a system to maintain a separate existence over time and to do it despite ongoing changes in the environment (even if these changes have not been foreseen) [4]. High relevance of availability of information in ensuring viability was emphasized already by S. Beer, the founder of a viable systems model – VSM [11].

The VSM reflects 5 interrelated functional systems (System 1 to System 5), each of which has unique role in the model (System 1 is operational system; other four systems are managerial ones). It is essential that each operational system can be a viable system itself, i.e., it can have a similar, composed of five functional systems, structure at a smaller scale of representation. So VSM has fractal architecture.

Often above-mentioned five systems are understood as organizational units [12]. However, VSM does not prescribe that each functional system corresponds to particular organizational unit. In general, the functions of Systems 1-5 can be performed by arbitrary actors (human actors, cross-departmental teams, software agents, etc.) belonging to arbitrary units of organizational structure.

Different communication channels exist between functional systems of VSM. These channels are reflected, analyzed, and illustrated in detail in [4]. Useful examples of content of information circulating in VSM are considered in [2], [3], and [4]. Different structural issues relevant to VSM are concerned in the research on fractal systems. These issues, such as similarity at several levels of scale, goal-orientation, self organization, and dynamics and vitality as well as the use of fractal paradigm in information systems development are discussed in, e.g., [8].

Information systems support for the VSM has not practically been discussed from the information systems engineering perspective in related work. This perspective requires clear distinction between information processing done by human actors and

information processing done by artificial systems. It also requires respecting variety of data, information, and knowledge flows in enterprises. This variety is analyzed in detail in [13]. Information flows from the role perspective are analyzed in research on information logistics, e.g., in [10]. This paper contributes (1) the generic information flow patterns based on VSM (Section 3), (2) the constructs for flow analysis represented in enterprise architecture description language (Section 4); and (3) the approach for use of the patterns and constructs (Section 4) that helps to identify and support (by means of IT) VSM compliant data, information, and knowledge flows.

3 Flows in VSM

In this section data, information, and knowledge flows in VSM are identified on the basis of VSM communication channels described in [4]. Here we do not yet distinguish between flow types, therefore we denote data, information, and knowledge flows by one general concept – DIK flows. DIK flow template is constructed using enterprise architecture description language ArchiMate 2.0 [9]. The flows are discussed solely from the functional perspective. This perspective is taken respecting the initial version of VSM [11] where it is emphasized that functions of the systems reflected in VSM can be differently distributed in enterprises. The flows are considered at one fractal level only and channels between different fractal levels of VSM [4] are not reflected to full extent in order to focus on DIK flows common for any fractal level.

System 1 (S1) in the VSM is responsible for the production and delivery of the enterprise’s goods or services to the pertinent environment. It may consist of several operational units (groups of functions) OU, which in turn consist of operational and managing units. Fig. 1 by dotted lines reflects DIK flows between the functions of the

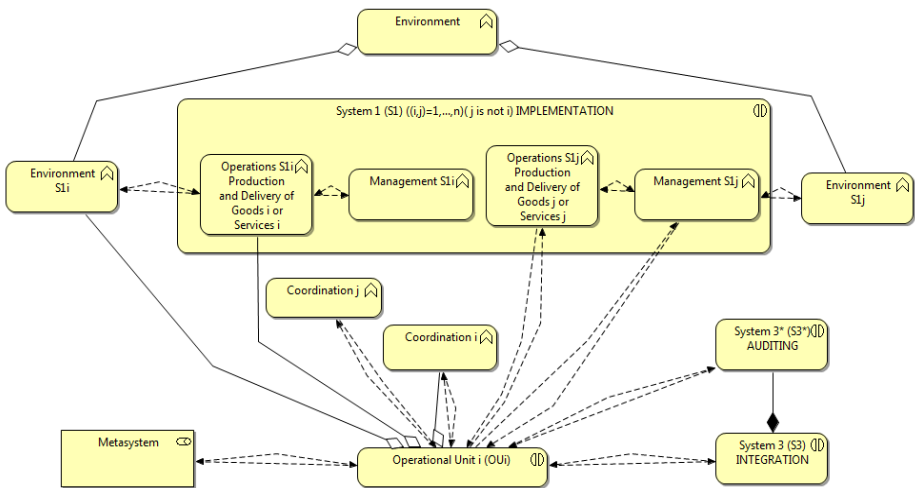


Fig. 1. DIK flow template for S1 (OU_i – shows interaction between the S1 and the environment). Similar templates can be created for S2, S3, S4, and S5.

S1. It shows a generic information flow template for S1. Similar templates can also be constructed for other systems of VSM.

Fig. 1 shows that functions of S1 are strongly dependent on various DIK flows inside and outside the system. According to [4] there are DIK flows that concern the Management System, i.e., System 3 (S3) and System S3* (S3*) for receiving instructions (S3 and S3* to OU_i), for accountability (OU_i to S3 and S3*), and resource bargaining; the flows to and from specific environment (including market of products and addressees of services of OU_i); the flows to and from regulatory units of OU_i from the System S2 (Coordination for OU_i and Coordination for other units OU_j ($j \neq i$) of S1); the DIK flows to and from other operational functions of S1; the flows to and from managerial functions of other operational units of S1; and the flows to and from the Metasystem (the next fractal level, if such exists). It is essential that all abovementioned units (except of the Metasystem) are represented as functions, i.e., they are not bound to be mirrored by structure of performers. This applies also to all other VSM systems, which are discussed in the remainder of this section.

Typical functions of *System 2* (S2) are associated with personnel policies, accounting policies, legal requirements, programming of production, organizational norms, etc. [4] Here the decisions concerning specific structure of S1 can be made. S2 receives and provides DIK flows from operational units of S1, it amalgamates DIK flows from all units and informs one unit about other units. It also filters and provides DIK flows for S3 as well as receives DIK flows from S3 and transmits them further to S1. In S2, there is a separate function for each operational unit of S1 and there is a corporate S2 function which directly relates to S3. It is important that S2 can change the structure of S1.

System 3 (S3) functions as operational enterprise management at a particular fractal level. There is also System 3* (S3*) that supports S3. The main function of S3* is ensuring the completeness of DIK flows, which reaches S3. S3 has to integrate operational units of S1, ensure that S1 functions harmoniously, and exploit synergies that might appear in S1. S3 assigns goals for each operational unit of S1 in cooperation with System 4 and in conformity with System 5 [4]. S3 is threefold supported by DIK flows which bring information about S1: directly from S1, via S2, and via S3*. This shows high importance of DIK flow completeness for managerial decisions. S3 has also DIK flows with System 4 and System 5. DIK flows also concern tasks where several management functions interact.

The principal responsibility of *System 4* (S4) is to identify and carry out, in a timely manner, internal changes necessary for the enterprise to remain viable. This system may exploit different tools for analyzing the environment and the impact of environmental changes on an enterprise (business intelligence tools, simulation, Delphi studies, etc.). The decisions made by S4 are produced in cooperation with S3 and under the approval of System 5. J.P. Rios [4] suggests to support the decision making environment by the space where current results of the enterprise's critical variables are displayed graphically and numerically; the space for providing DIK about enterprise's history, the space for displaying simulation model, the space for visualization of the VSM, as well as the space for showing static and dynamic images relating to decision-making and web access via DIK flows inside and outside the enterprise.

Functions of *System 5* (S5) possess the highest authority in the enterprise. S5 functions must balance the present and future of the enterprise, establish and maintain the “identity” of the enterprise by, e.g., determining vision, mission, and strategic goals of the enterprise. S5 has DIK flows to and from S3 and S4.

The most complex DIK flows are within and around S1. Here we have to take into consideration that the S1 itself, being a viable system, may include all information flows relevant to S2, S3, S4, and S5 at another fractal level. One more issue to be taken into consideration is the necessity to have information system support for each interaction of several systems. Specific interaction facilities have to be available to ensure needed exchange of DIK flows.

The role of DIK flows, IT, and information systems is ambiguously reflected in the related work on VSM, e.g., [4] reports about tools for the visualization and use of communication channels of VSM. In [4] the information system concept is related, on one hand, to S2 and, on another hand, to S3*, simultaneously stating the need for IT support for S4 without systemic view on enterprise information systems. In this paper we look at VSM from information systems engineering perspective. DIK flows and proposed templates can help to identify a set of information flows that are mandatory for enterprise to comply with VSM. To help to identify needed IT support for each flow, we have developed specific constructs that are described in the next section.

4 Flow Analysis Constructs

In this section we examine in more detail different types of flows. A flow can exist between two data processing nodes of any nature. Information is data interpreted by knowledge belonging to a node. Knowledge of the node may be changed on the basis of information obtained via data processing [13]. Thus, in this paper, we define that *data flow* exists either, when information and knowledge flow exists, or in cases, when node that receives the flow has no interpretation function. *Information flow* exists if data is interpreted by knowledge of the node that receives the flow (regardless whether the interpretation function of knowledge is or is not changed). *Knowledge flow* exists when data, after the interpretation, causes the changes in the interpretation function of the node.

The basic architectural construct proposed for analysis of the flows is shown in Fig. 2. It is represented by enterprise architecture (EA) description language ArchiMate 2.0 [9] and consists of three levels. The construct describes how the flow between two data processing nodes can be implemented. At the technology level the data processing element is Node; at the application level – Application Component; at the business level – Actor. For each piece of information there is an interpretation created by an actor according to the knowledge the actor possesses. By interpreting information, the actor’s knowledge creates the meaning [13] which can be transferred to other actors or to applications. The applications are bound to use “approved meanings”, which correspond to organizational ontology, meta-data, or other codified common organizational knowledge [14].

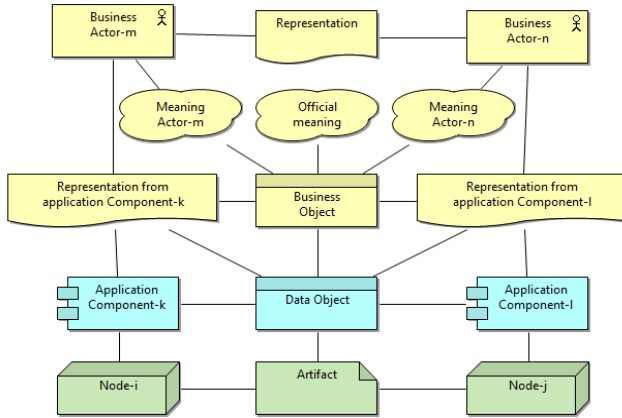


Fig. 2. Generic enterprise architecture construct used for flow analysis

The flow construct shows three Representations – one human and two application made ones, as well as three Meanings, the one for actor on the left, the one for actor on the right, and the official meaning. The construct can illustrate data, information, and knowledge flows from the elements on the left side to the elements on the right side. *Knowledge flow* is the most complex concept. We assume that knowledge flow incorporates data and/or information flows, which optionally may be supported by IT solutions. The knowledge flow will appear in the case of propagation of changes in Meaning elements. The Meaning may change because of changes in explicit or mental models used by particular actors or because of changes in models used by the enterprise in general [14]. Different variations of a knowledge flow are possible, e.g., there can be knowledge flows that are not supported by IT applications (e.g., face-to-face communication of two or more actors), there can be flows where official meanings are not changed, only actors’ individual knowledge (meaning) changes; there can be knowledge flows where enterprise data non-related software and hardware are used for transferring the data, e.g., by sending an email message; there can be cases when the official meaning is changed (e.g. A in Fig. 3) and propagated via different paths in the construct (only one propagation path is shown in A part of Fig. 3).

There is a particular type of construct corresponding to each knowledge flow. Additionally, there are specific constructs for *information flows* that are not part of knowledge flows (B in Fig. 3). These information flows do not change content of Meaning, however, the Meaning has to exist for data processing nodes to be able to interpret the data. In Fig. 3 part B the links to official meanings are shown by bold lines. The information occurs only if there is knowledge that can interpret it. The interpretation may vary with respect to its correspondence to official meaning, i.e., the same data can cause different information to be perceived by nodes that interpret it, e.g., one Actor may interpret data as it is expected by the official meaning, but

another actor can interpret it differently. *Data flow* is behind each knowledge and information flow. Usually data flows have to be considered only in the context of knowledge and information flows.

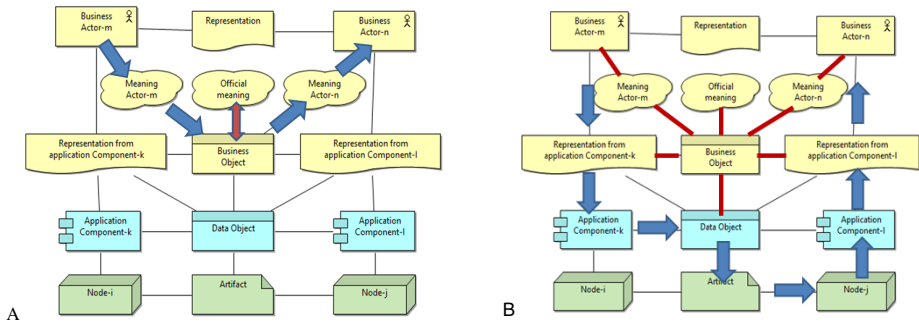


Fig. 3. Two types of constructs A) knowledge flows without using application software; B) information flow using application software run on Node-j

To ensure that all information flows prescribed by VSM are present in an information system the following approach is proposed:

- 1) Examine each generic DIK flow in each VSM based flow template (an example of the template for System 1 is shown in Fig. 1).
- 2) For each identified generic flow, identify all flow instances.
- 3) Analyze each flow instance using the generic enterprise architecture construct (Fig. 2) and develop the exact construct types (Fig. 3) for each flow instance.
- 4) Check whether all constructs corresponding to the identified flow instances can be integrated into the information system.

5 Conclusions

While VSM is a well-known model for achieving viability of enterprises, in related research it has not been analyzed from information systems engineering perspective. The research presented in this paper shows that it is possible to use VSM in information systems engineering. The paper proposes (1) the data, information, and knowledge (DIK) flow templates, (2) the constructs in enterprise architecture description language, and (3) the approach of use of the aforementioned templates and constructs. The proposed approach can help to ensure that all flows prescribed by VSM are present in an enterprise information system.

Further work on this topic concerns (1) controlled experiments with the templates and the constructs, (2) formalization of flow analysis algorithms, and (3) developing software tools that can support the approach reflected in this paper.

Acknowledgment. The work on this paper was supported by Latvian National Research Programme 2, in particular, Project 5.

References

1. Missikoff, M., Charabilidis, Y., Gongcalves, R., Popplewell, K. (eds.): *FINES Research Roadmap 2025: Final Document (Version 3.0)*, European Communities (2012), http://cordis.europa.eu/fp7/ict/enet/documents/fines-research-roadmap-v30_en.pdf
2. Hoverstadt, P.: *The Fractal Organization: Creating Sustainable Organizations with the Viable Systems Model*. John Wiley & Sons, Chichester (2008)
3. Espejo, R., Reyes, A.: *Organizational Systems. Managing Complexity with the Viable System Model*. Springer, Berlin (2011)
4. Perez Rios, J.: *Design and Diagnosis for Sustainable Organization*. Springer, Berlin (2012)
5. Sandkuhl, K., Kirikova, M.: Analysing enterprise models from a fractal organisation perspective - potentials and limitations. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) *PoEM 2011*. LNBIP, vol. 92, pp. 193–207. Springer, Heidelberg (2011)
6. Strazdina, R., Kirikova, M.: Change management for fractal enterprises. In: *Proceeding of the 19th International Conference on Information Systems Development (ISD 2010)*, pp. 735–745. Springer, Prague (2011)
7. Kirikova, M.: Towards flexible information architecture for fractal information systems. In: Kusiak, A., Lee, S. (eds.) *The Proceedings of the International Conference on Information, Process, and Knowledge Management, eKNOW 2009, Cancun*, pp. 135–140. IEEE Computer Society (2009)
8. Kirikova, M.: Towards multifractal approach in IS development. In: Barry, C., Conboy, K., Lang, M., Wojtkowski, G., Wojtkowski, W. (eds.) *Information Systems Development: Challenges in Practice, Theory and Education*, vol. 1, pp. 295–306. Springer (2009)
9. ArchiMate: ArchiMate® 2.0 Specification (2012), <https://www2.opengroup.org/ogsys/catalog/c118>
10. Sandkuhl, K., Smirnov, A., Shilov, N.: Information logistics in engineering change management: Integrating demand patterns and recommendation systems. In: Niedrite, L., Strazdina, R., Wangler, B. (eds.) *BIR 2011 Workshops*. LNBIP, vol. 106, pp. 14–25. Springer, Heidelberg (2012)
11. Beer, S.: *Diagnosing the Systems for Organizations*. John Wiley & Sons, Chichester (1985)
12. Kontogiannis, T., Malakis, S.: A systemic analysis of patterns of organizational breakdowns in accidents: A case from Helicopter Emergency Medical Service (HEMS) operations. In: *Reliability Engineering and System Safety*, vol. 99, pp. 193–208. Elsevier (2012)
13. Rudzajs, P., Kirikova, M.: Multimodal information logistics for conceptual correspondence monitoring. In: *The Proceedings of the 5th Workshop on Information Logistics (ILOG 2012) in conjunction with the 11th International Conference on Business Informatics Research (BIR 2012)*, Russia, Nizhniy Novgorod (2012)
14. Kirikova, M.: Domain Modeling Approaches in IS Engineering. In: Osis, J., Asnina, E. (eds.) *Model-Driven Domain Analysis and Software Development: Architectures and Functions*. IGI Global (2010)

On Materializing Paths for Faster Recursive Querying

Aleksandra Boniewicz, Piotr Wiśniewski*, and Krzysztof Stencel

Faculty of Mathematics and Computer Science, Nicolaus Copernicus University,
Toruń, Poland

`grusia,pikonrad,stencel@mat.umk.pl`

Abstract. Recursive data structures are often used in business applications. They store data on e.g. corporate hierarchies, product categories and bill-of-material. Therefore, recursive queries as introduced by SQL:1999 or formerly implemented by Oracle constitute a useful facility for application programmers. Unfortunately, recursive queries are not implemented by a number of database systems with MySQL as the most profound example. If an application has such a database as the backend storage, recursive queries will be usually hard-coded at the client side. This is not efficient. In this paper we propose using redundant data structures to answer recursive queries quicker. Such structures must be synchronized in response to updates of data. This means a significant processing overhead for updates. We present experimental evaluation to show losses and gains caused by our solution for various usage scenarios. They prove the feasibility of our proposal. We show our proof-of-concept implementation as a part of the Hibernate framework. Thus, application programmers are separated from all internals of necessary database objects and triggers. These are created automatically by Hibernate generators.

1 Introduction

Recursive and hierarchical database structures are common in business applications. They are used to represent corporate hierarchies, various classifications in e.g. libraries and shops, bills of material, road networks etc. There exists numerous ways to persist them [1]. There is also a well-recognised business need to query such structures efficiently and conveniently. In 1985 Oracle introduced its handy CONNECT BY query construct. However, it has never become a part of the SQL standard. Another paradigm was voted into SQL:1999 based on recursive Common Table Expressions. It has been implemented in numerous database systems [2], while the academia worked on optimization methods for such queries [3,4,5]. However, there are still database systems whose SQL dialect does not include any recursion in queries. The most notable example of such a database system is MySQL. It frequently plays the role of the backend storage in web

* Supported by the Polish National Science Centre grants 2011/01/B/ST6/03867.

applications. In such projects developers have to hardcode the recursive queries into the logic of the application. On the other hand, researchers and practitioners should not limit themselves to the actual implementation of recursive queries if a given DBMS provides them. Potentially, there could be faster ways to query hierarchies and networks.

Object-relational mapping systems (ORM) [6] constitute a tempting opportunity to pursue the answers to the abovementioned questions. Originally, they were to bridge the gap between relational storage and object-oriented code known under collective name *impedance mismatch*. Of course they do [7,8], but since they provide an object-oriented abstraction layer, they can be used to introduce additional logic separated from application programmers. In our research, we exploit this opportunity. We have prepared proof-of-concept augments to Hibernate ORM that realize recursive queries [9] and partial aggregation [10]. In particular, we experimented with adding recursion on top of database systems that do not implement it directly [11].

In this paper we describe another solution to this problem. We propose extending ORM with other redundant data structures that facilitate recursive queries. This ORM extension allows posing recursive queries, especially when the back-end database does not implement them. Our solution obviously is also an option for a DBMS that has SQL:1999 recursion, but the architect does not want rely on it. We have prepared a proof-of-concept implementation of this ORM extension and conducted experiments with the efficiency with respect to the state-of-the-art research.

Since our solution is based on materializing redundant data, its efficiency depends on the usage scenario of the application. Updates of stored data must be reflected in the materialized derived data. Therefore, we experiment with the gains of efficiency of queries and the overheads imposed on updates. If the update/query ratio is similar to that of common applications, the proposed solution will provide significant profits. However, if updates dominate, our proposal will slow down the application and should not be used.

The contributions of this paper are as follows:

- a proposal to use redundant materialized data to accelerate recursive queries without SQL:1999 recursive database facilities,
- a design of an extension to Hibernate that allows running direct SQL:1999 recursive queries, even if the underlying DBMS does not support them,
- a proof-of-concept implementation of this extension.

The paper is organized as follows. In Section 2 we address the related work. In Section 3 we present the database structures and the method to run recursive queries without native DBMS recursion. Section 4 reports on the experimental evaluation of our proposal. Section 5 concludes.

2 Related Work

In relational databases hierarchical data is usually represented in a table having a foreign key that references the same table. The other option is to use two

empId	fname	sname	bossId
1	John	Travolta	
2	Bruce	Willis	1
3	Marilyn	Monroe	
4	Angelina	Jolie	3
5	Brad	Pitt	4
6	Hugh	Grant	4
7	Colin	Firth	3
8	Keira	Knightley	6
9	Sean	Connery	1
10	Pierce	Brosnan	3
...			

Fig. 1. Sample data on the corporate hierarchy for multilevel marketing

tables. The first contains vertices, while the second stores edges. Such a solution means a better flexibility at the cost of reduced efficiency of queries. Execution and optimisation of recursive queries is a topic of ongoing research [3,4]. Direct features to pose recursive queries were introduced into SQL yet in 1999. Until then only a few DBMSs had implemented such querying facilities. A survey on the implementations of SQL:1999 recursion can be found in [2].

The simplest recursive data structure is a hierarchy stored in tables with self-referencing foreign key. In this paper we use a corporate hierarchy as a running example. We assume that the company business model is based on multilevel marketing. The depth of the hierarchy tree in such a firm is not limited by any prescribed depth. It depends on the smartness of individuals. Figure 1 shows sample data of a database table `emp`.

In prequel research we presented how to integrate recursive queries with object-relational mapping systems [9]. In particular, we showed a convenient API for Hibernate that allows defining recursive queries by simple annotations to Java entity classes.

Unfortunately, still there are database systems that do not implement recursive queries in any form. One of the most popular of them is MySQL. It is frequently used within the LAMP paradigm in web applications. Such applications usually contain hierarchical data, e.g. product categorization in shops or hierarchy of posts in web forums. In order to cater for the needs of their developers, we designed and implemented a methodology to run such queries efficiently even with MySQL as the backend storage [11]. We integrated this implementation with Hibernate. Thus, programmers have a uniform interface regardless of the chosen DBMS.

We presented two methods to query the recursive structures without using recursive queries. Both have been integrated with API [9]. The first method is called *horizontal unrolling*. It joins the base table `maxlevel` times.

The second method is called *vertical unrolling*. It uses temporary tables and consists in sending several queries and processing partial answers. Both methods

baseId	upId	distance
8	6	1
8	4	2
8	3	3

Fig. 2. Tuples on the path to the root from the tuple (8, 'Keira', 'Knightley', 6)

are significantly faster than the naïve method that in a loop sends a separate query for each visited node. From these two methods the horizontal unrolling seems to be more efficient.

The method of unrolling (either horizontal or vertical) is indicated by the parameter `method` in the annotation `@unrolling(unrolling method)`.

3 Materialization of Paths

In this paper we propose using redundant data to optimize the queries to hierarchies. We store full paths from each node to the root of the tree that contains this node. Each path is represented as a number of tuples in the table `Paths`. We call this materialization method *full paths* unrolling. An example of materialized redundant data is presented on Figure 2.

If an application programmer wants to use the full paths unrolling, the entity class will be annotated `@unrolling(method = "full paths")`. Then at the first call to the database the table `Paths` is created. After the table `Paths` is populated, appropriate triggers on the table `Emp` are installed. Their duty is to keep paths up to date. If the backend database has no implementation of recursive queries, Hibernate will call the query presented on Listing 1 in response to the call of the method `getRecursive(String)`.

Listing 1. Query sent by `getRecursive("Travolta")`

```
SELECT e.empId, e.fname, e.sname, e.bossId
FROM emp b JOIN path p ON (p.UpId = b.empId)
      JOIN emp e ON (p.baseId = e.empId)
WHERE b.sname = 'Travolta'
```

4 Experimental Performance Evaluation

We performed tests on MySQL installed on a computer with AMD Phenom II 3,4GHz, 8GB RAM and 2 Caviar Black 7400rpm 500 GB HDDs. We have chosen MySQL, since it is probably the most popular database without any implementation of recursive queries. The database has been installed as the default configuration without any tuning tricks.

We populated the base table `Emp` (see Figures 1) with data of different volumes and levels of hierarchy. For each volume, we generated data with 5, 10, 15 and 20

Table 1. Times required to build the initial materialization of paths

	5 levels	10 levels	15 levels	20 levels
1 000	01.62	01.18	01.20	01.22
10 000	03.85	06.34	08.41	15.73
100 000	43.68	01:04.82	01:25.28	02:10.72
1 000 000	08:22.51	25:05.28	46:52.04	01:23:03.00

levels of hierarchy. The volumes were 10^3 , 10^4 , 10^5 and 10^6 records. Therefore, altogether we tested our solution on 16 data sets.

In the tests we compared the full paths unrolling proposed in this paper (marked with by FP in tables) against the horizontal unrolling (denoted by H-UN). We did not take into account the vertical unrolling since it was always slower than the horizontal [11]. The tables present the execution times of queries implementing both methods and the ratio between the FP runtime and H-UN runtime.

4.1 Building the Materialization of Paths

In the first test we measure only the full paths unrolling, namely we assess the overhead caused by the initial population of the derived table.

Assume that we connect Hibernate to a database that already has a non-empty table `Emp` and we plan to employ the full paths unrolling. Therefore, we have to populate the table `Paths` with necessary materializations. The size of this table depends on the size of `Emp` and the number of hierarchy levels. Table 1 presents the results of this experiment for all 16 database settings. For smallest tables populating the table `Paths` takes less than 2 seconds. The most complex case (one million records and 20 levels) requires over one hour of initial computation.

4.2 Retrieving a Subtree

In this test we measured the runtime of subtree retrieval for a given node. The experiment consisted in retrieving the subtree of a randomly chosen node at a randomly chosen level. For FP and H-UN the same node have been used each time. The experiment has been repeated 20 times. Table 2 presents the results of this experiment.

There is an interesting phenomenon in these results. If the `Emp` table contains 100 000 records, and the trees are not too deep (5 or 10 levels), the full paths unrolling is significantly slower than the horizontal unrolling. We observed this phenomenon in most runs of the test. Moreover, we repeated this tests on a different server with a totally different hardware configuration and the default configuration of MySQL. The phenomenon reoccurred. In our opinion, it is a consequence of the semantics of the database memory pool. The computation for the full paths unrolling operates on two tables instead of one. Therefore for some volumes, the computations for FP spills out of the database cache. On the

Table 2. Times to retrieve the tree spanned by a random node

	5 levels			15 levels			20 levels		
	FP	H-UN	RATIO	FP	H-UN	RATIO	FP	H-UN	RATIO
1 000	00.06	00.09	66.67%	00.05	00.06	83.33%	00.08	00.31	25.81%
10 000	00.10	00.14	71.43%	00.10	00.11	90.91%	00.21	02.48	8.47%
100 000	00.73	00.70	104.29%	00.18	00.19	94.74%	00.30	24.63	1.22%
1 000 000	01.09	01.51	72.19%	01.72	03.13	54.95%	21.18	18:12.84	1.94%

Table 3. Times to retrieve the root for a random node

	10 levels			20 levels		
	FP	H-UN	RATIO	FP	H-UN	RATIO
100 000	00.37	00.69	53.62%	00.17	24.63	0.69%
1 000 000	00.92	02.68	34.33%	08.81	18:12.84	0.81%

other hand, the default database cache is enough for the horizontal unrolling in this setting.

4.3 Finding the Root for a Node

In this test we queried for the root of a random node. Table 3 presents the result of this experiment for bigger databases and hierarchy depths 10 and 20. The results exhibit the intuitive tendency.

4.4 Synchronization

Database optimizations using redundant materialization require cautious consideration of the overhead caused by the synchronization of derived data at each modification of base data. Table 4 shows the measures of this overhead for operations that change the structure of trees. Obviously, we measure only FP, since H-UN has no such overhead. The results of experiments prove that the overhead is moderate. Thus, the full paths unrolling is feasible for practical applications. We elaborate on it in the next Section.

4.5 Balancing Updates and Queries

Finally, we show the gains and losses caused by the proposed method of full paths unrolling. Each test run contained 100 operations. The percentage header indicates how many of them were updates. The tests were performed with databases having 100 000 and 1 000 000 records. In both tests the trees have 20 levels. Table 5 presents the results of this test. Under query/update ratios typical to numerous applications, the full paths unrolling outperforms the horizontal method.

Table 4. Times to synchronize the materialization for a single update of a base row

	5 levels	10 levels	15 levels	20 levels
1 000	00.10	00.14	00.22	00.37
10 000	00.24	00.51	00.97	01.73
100 000	01.55	04.44	08.75	16.24
1 000 000	15.59	46.15	01:29.02	06:24.13

Table 5. The performance of both unrolling methods under various ratios of queries and updates in an application

	5%			10%			20%		
	FP	H-UN	%	FP	H-UN	%	FP	H-UN	%
100 000	00:19.03	00:27.34	70%	00:44.14	00:27.07	163%	01:00.12	00:27.49	219%
1 000 000	01:45.94	04:46.64	37%	03:54.83	04:44.81	82%	09:22.22	04:43.75	198%

5 Conclusion

In this paper we have discussed a method to run SQL:1999 recursive queries against database systems that do not implement them. The presented method called *full paths unrolling* has amounted to be considerably efficient in case of (1) deep hierarchies and (2) shallow hierarchies and large number of rows. For other experimental settings the method is comparable to the horizontal unrolling. The results of queries run under all these methods are correct, i.e. the same as the results of the equivalent SQL:1999 query using the clause `WITH RECURSIVE`.

The disadvantage of full paths unrolling is inherent in its redundant materialized data structures. Any update to the original data causes a corresponding modification of the derived path data. However, if the ratio of updates and queries is similar to the common situation in production applications, the full paths unrolling will be superior to other methods. On the other hand, if updates clearly dominate, even the naïve looping solution can be the best option.

We have developed our proof-of-concept implementation of the new method as a part of Hibernate ORM. As the result, we have separated application programmers from all technicalities of the solution. They can conveniently use recursive queries to the hierarchical data whether the backend database system has SQL:1999 recursion or not.

References

1. Brandon, D.: Recursive database structures. *J. Comp. Sci. Coll.* 21, 295–304 (2005)
2. Przymus, P., Boniewicz, A., Burzańska, M., Stencel, K.: Recursive query facilities in relational databases: A survey. In: *FGIT-DTA/BSBT*, pp. 89–99 (2010)
3. Ordonez, C.: Optimization of linear recursive queries in SQL. *IEEE Trans. Knowl. Data Eng.* 22, 264–277 (2010)
4. Burzańska, M., Stencel, K., Wiśniewski, P.: Pushing predicates into recursive SQL common table expressions. In: *Grundspenkis, J., Morzy, T., Vossen, G. (eds.) ADBIS 2009. LNCS, vol. 5739, pp. 194–205. Springer, Heidelberg (2009)*

5. Cortesi, A., Halder, R.: Abstract interpretation of recursive queries. In: Hota, C., Srimani, P.K. (eds.) ICDCIT 2013. LNCS, vol. 7753, pp. 157–170. Springer, Heidelberg (2013)
6. Melnik, S., Adya, A., Bernstein, P.A.: Compiling mappings to bridge applications and databases. *ACM Trans. Database Syst.* 33 (2008)
7. O’Neil, E.J.: Object/relational mapping 2008: Hibernate and the Entity Data Model (EDM). In: Wang, J.T.L. (ed.) SIGMOD Conference, pp. 1351–1356. ACM (2008)
8. Bauer, C., King, G.: Java Persistence with Hibernate. Manning Publications Co., Greenwich (2006)
9. Wiśniewski, P., Szumowska, A., Burzańska, M., Boniewicz, A.: Hibernate the recursive queries - defining the recursive queries using Hibernate ORM. In: ADBIS (2), pp. 190–199 (2011)
10. Gawarkiewicz, M., Wiśniewski, P.: Partial aggregation using hibernate. In: Kim, T.-H., Adeli, H., Slezak, D., Sandnes, F.E., Song, X., Chung, K.-I., Arnett, K.P. (eds.) FGIT 2011. LNCS, vol. 7105, pp. 90–99. Springer, Heidelberg (2011)
11. Boniewicz, A., Stencel, K., Wiśniewski, P.: Unrolling SQL: 1999 recursive queries. In: Kim, T.-H., Ma, J., Fang, W.-C., Zhang, Y., Cuzzocrea, A. (eds.) EL/DTA/UNESST 2012. CCIS, vol. 352, pp. 345–354. Springer, Heidelberg (2012)

XSLTMark II – A Simple, Extensible and Portable XSLT Benchmark[★]

Viktor Mašíček and Irena Holubová (Mlýnková)

Department of Software Engineering, Charles University in Prague, Czech Republic
viktor@masicek.net, holubova@ksi.mff.cuni.cz

Abstract. In this paper we focus on the problem of XSLT benchmarking. Although it is a straightforward task, currently there exists only a single XSLT benchmark which is obsolete and no longer supported. Hence we have proposed a novel tool called *XSLTMark II* having several important features such as simplicity, portability, extensibility, and wide parametrization. It allows for generating of test cases from templates of tests, running tests, generating XML reports, transforming reports into HTML format and testing different XSLT processors. The basic set of templates was created on the basis of analysis of real-world XSLT scripts. And, last but not least, a proof of the concept is provided via application of the benchmark on a selected set of XSLT processors.

1 Introduction

A *benchmark* or a *test suite* is a set of testing scenarios or test cases, i.e. data and related operations which enable one to compare versatility, efficiency or behavior of the *system(s) under test*. In our case the set of data involves XML documents, whereas the set of operations can involve any kind of XML-related data operations. The technology we want to focus on in this paper is XSLT [18], i.e. templates that describe the way the given XML document should be transformed to another (text) output.

Although the problem of benchmarking of XSLT processors is a natural and straightforward task [16], currently there exists only a single XSLT benchmark which is obsolete and no longer supported [11]. Hence we have proposed a novel tool called *XSLTMark II*. It has several important features such as simplicity, portability, extensibility, and parametrization. It allows for generating of test cases from templates of tests, running tests, generating XML reports, transforming reports into HTML format and testing different types of XSLT processors. The basic set of templates was created on the basis of analysis of real-world XSLT scripts. And, last but not least, a proof of the concept is provided via application of the benchmark on a selected set of XSLT processors.

The paper is structured as follows: In Section 2 we discuss the related work. In Section 3 we describe the problem of XSLT benchmarking and the way we have created *XSLTMark II*. In Section 4 we provide a set of results of preliminary tests using the benchmark and in Section 5 we conclude.

[★] Supported by the grant SVV-2013-267312.

2 Related Work

Considering XML technologies, the set of available benchmarks [16] involves a number of query benchmarks covering XQuery (*XMark*, *XOO7*, *XMach-1*, *MBench*, *XBench*), XPath (*XPathMark*), related transaction processing (*TPoX*) and a general repository for query benchmarks (*MemBeR*). However, regarding the area of XSLT benchmarks, the situation is much worse.

XSLTMark [11] is one of the best known XSLT benchmarks, whereas other benchmarks (e.g. *Caucho* or *David Parshley*) are based on it. Unfortunately, it is outdated. The published results are from 2001, it is no longer maintained and it is not downloadable now. It contains 40 synthetic tests and it has good metrics for measurement of transformation speed. It uses kilobytes-per-second value for each test, where kilobytes are the average of input and output document size. The result for one XSLT processor is the sum of kilobytes-per-second value for all tests.

XSLTMark rates only speed and correctness. Unfortunately, it does not cover other criteria, such as documentation and manageability, that can be important for some consumers. It is possible to upgrade XSLTMark with new tests and to add new tested processors. However, upgrades are no longer possible, because it is not downloadable at this moment.

3 XSLT Benchmarking and XSLTMark II

In general, XSLT processors exist as separate *programs* (either allowing only XSLT transformations or also other functions), downloadable *libraries* for programming or scripting languages (Java, C++, PHP etc.), or components of web *browsers*. Some processors can be naturally classified into multiple such types. The processors we used for testing to show the capabilities of *XSLTMark II* are Saxon [13], Xalan [4], XT [5], libxslt [6], MSXML [3], and Sablotron [17]. Our aim was to create a benchmark which enables testing of speed, correctness and memory usage, allowing to test different types of XSLT processors, repeated set up tests, extensibility of test sets and addition of new XSLT processors. Our first step was to prepare the basic set of test cases with regard to XSLT documents used in real-world applications. For this purpose we collected and analyzed 5,787 XSLT files for frequency of XSLT constructs, depth of XML tree, version of XSLT, fan-out of elements etc. Similarly, we prepared a list of typical categories of used documents (e.g. *RSS* [7] or *DocBook* [19]) having their specifics representing typical applications.

In the next step we implemented the core testing program, *XSLTMark II*, that allows the user to add new configurable tests, testing of different XSLT processors and reporting the results. It can be run on different operating systems and modified using various settings. On the basis of the previous analysis, we created a basic (core) set of testing scenarios which covers the typical use cases. Most of them were created configurable for future extension.

3.1 Collecting and Analysis of Real-World XSLT Scripts

For the purpose of downloading a representative set of XSLT scripts, we used the *Wget* [12] crawler. We used various methods for searching the data, mainly *Google Search* [1] and *Google Code* [2], and finally we downloaded 19,650 XSLT files. We denote these data as *dirty*, since there were duplicities of documents or some non-XSLT documents. Consequently, we had to merge all the data downloaded by all methods, correct their contents, remove non-XSLT files, duplicities and similarities. Finally, we collected 5,787 documents for analysis after such *cleaning*.

Next, we analyzed the complexity of their content and identified categories of documents. The key features and findings are as follows:

- **Maximum Depth:** The files with maximum depth between 0 and 20 represented 99% of the analyzed files.
- **Depth of Nesting:** Constructs of XSLT language `for-each`, `choose` and `if` increase the complexity of XSLT stylesheets (as similar constructs in other programming languages). We found out that more than 99% of files have maximum depth of nesting of all selected elements less than or equal to 5.
- **XSLT Version:** About 85% of analyzed files have version 1.0, about 7% version 2.0.
- **Fan-out of Elements:** 98% of files have the average fan-out of 1. 86% of the analyzed files have maximum fan-out less than 40.
- **Element Numbers:** W3C XSLT specification defines many elements in namespace `xs1`. However, not all of them are used in practice. An important finding is that 99% elements are from XSLT version 1.0, whereas about 95% of all elements are the same 16 elements.
- **Size of Files:** The size of a file is a very important feature. Some XSLT processors may have a problem with big XSLT templates due to lack of the memory. However, about 74% of analyzed files are smaller than 10kB and 96% files are smaller than 50kB.
- **Recursion:** Named templates in XSLT (element `template` with attribute `name`) can be understood as functions. Thus, we wanted to find their recursive calls in analyzed files. We found out that 87% of the analyzed files do not involve recursion and 7% have only 1 recursion cycle. Moreover, the longest recursion cycle in files has length 1 (the template calls itself) in 95%.
- **Output Format:** In more than 50% of all analyzed files there is no pre-set output format. 87% of these files have default output format `XML`.

After scanning all files and describing all their main features, we detected whether they fit into one of the predetermined categories. The categories were selected on the basis of another research of the Internet and typical XSLT applications. For each file we found the number of occurrences of properties from the list created for each category. The sum of the numbers of occurrences, multiplied by a weight assigned to specific property, is called *Property Value* (PV).

$$PV = \sum_{properties} (\# \text{ occurrences of property}) \times (\text{weight of property}) \quad (1)$$

The PVs were then compared to threshold values that were set for each category. The lists of properties and threshold values for each category were set by manual research and repeated data scan and correction. Due to space limitations we refer the reader interested in particular criteria and threshold values to [15]. The numbers of files classified into selected categories are provided in Table 1.

Table 1. Number of files belonging to categories

	Category	#	%
1	<i>RSS reader</i> – RSS-to-XML or RSS-to-HTML transformation	81	1.40%
	<i>RSS generator</i> – XML-to-RSS transformations	30	0.52%
2	<i>Google Search Appliance</i> – transformation of layout of the search result	13	0.22%
3	<i>GraphML</i> – generation and transformation of graphs	4	0.07%
4	<i>XGML</i> – graph description based on GML	8	0.14%
	<i>LOGML</i> – representation of the structure of the Web in Web servers	0	0.00%
5	<i>DocBook reader</i> – DocBook-to-XML/HTML/PDF transformation	719	12.42%
	<i>DocBook generator</i> – XML-to-DocBook transformation	1	0.02%
6	<i>RDF reader</i> – RDF-to-XML/HTML/PDF transformation	18	0.31%
	<i>RDF generator</i> – XML-to-RDF transformation	141	2.44%
	<i>RDFS reader</i> – RDFS-to-XML/HTML/PDF transformation	6	0.10%
	<i>RDFS generator</i> – XML-to-RDFS transformation	1	0.02%
	<i>RGML reader</i> – RGML-to-XML/HTML/PDF transformation	0	0.00%
	<i>RGML generator</i> – XML-to-RGML transformation	0	0.00%

3.2 Test Environment

Our aim was to create a test environment which enables running of parameterized test cases for different XSLT processors, reporting the results, simple adding of new test cases, adding more XSLT processors, running on different operating systems and monitoring of time and memory usage. We implemented all the requirements in program called *XSLTMark II* available at [9]. Most of the functionality was implemented using drivers, so it is easy to add new features by just adding a new driver with distinct interface. For example, the generator of test cases uses prearranged *Smarty* [14] and *ToXgene* [10] generator drivers and adding of new generator drivers is possible and very easy.

We chose PHP as a language for the implementation. The main advantage is easy portability between OSs and ability to run from console. Moreover, it is easy to add an extension for running from a Web browser.

3.3 Test Cases

One *test case* includes one XSLT template and a set of couples of input and expected files. Thus, quaternion *processor*, *XSLT template*, *input*, *expected output*

determines one record in a *report*. Possible generated errors are written into reports as well. The output of generated transformation is compared with the expected output. The result of the comparison is written into the reports too. As we have mentioned, *XSLTMark II* is fully extensible. However, it is provided with a basic set of test cases. Their full description can be found in [15].

Some XSLT templates were designed as synthetic and some XSLT templates could be implemented more efficiently. We also created some tests based on real-world usage and, at the same time, we used test cases that correspond to real-world XSLT templates used in practice.

4 Preliminary Tests

To demonstrate the usability of *XSLTMark II* we performed a set of basic tests using the default set of test cases. We compared different versions of processors, different types of processors (e.g. library and program), average time and memory usages (see Figures 1 and 2), running of tests on different OSs etc.

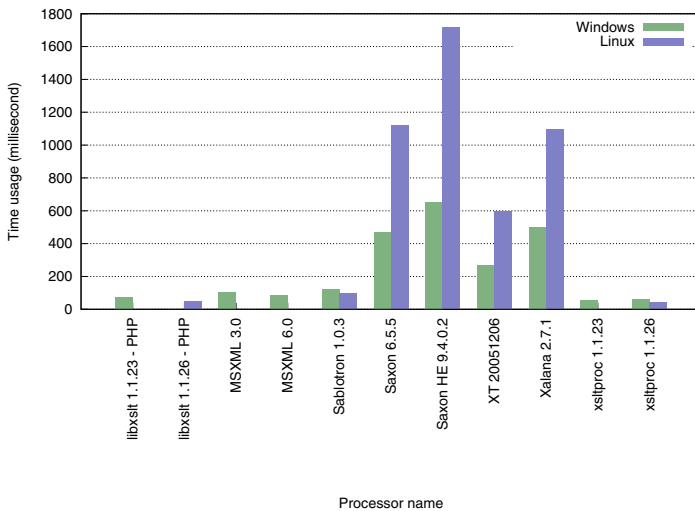


Fig. 1. Average time usages for processors

A summary of all discussed features for all tested processors is provided in Table 2. (Complete reports of tests are available at [9].) Here is the list of features with their short explanations:

1. **Ver.** – The maximum supported XSLT version.
2. **Enc.** – The list of supported encodings. “All” means all tested encodings.
3. **Speed** – The speed of a processor in a verbal expression.
4. **Mem.** – The memory usage of a processor in a verbal expression.

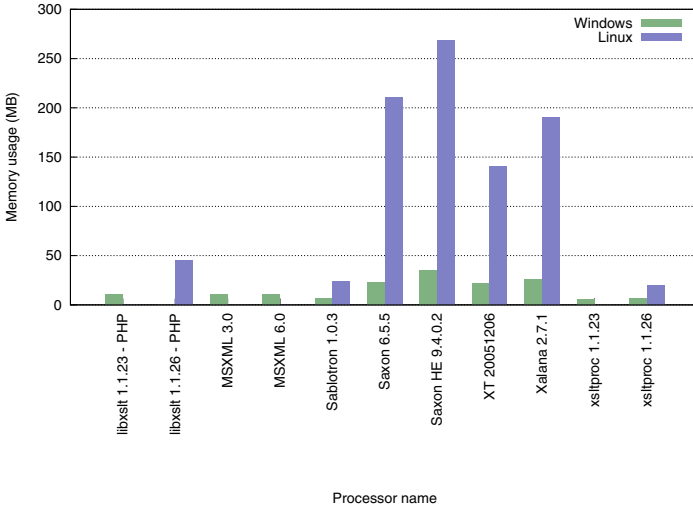


Fig. 2. Average memory usages for processors

5. **Ind.** – A flag indicating whether an indented XSLT template is better than a non-indent XSLT template.
6. **Proc.** – A flag indicating whether procedural or non-procedural approach is better for speed of a processor.
7. **Err.** – The number of failed tests from all 43 test cases.
8. **Java** – A flag indicating supporting of Java functions.
9. **DB** – A flag indicating the test case of category DocBook.
10. **NS** – A flag indicating the tests of namespace aliasing.

Table 2. The list of tested processors with their discussed features

Processor	Ver.	Enc.	Speed	Mem.	Java	DB	Err.	NS	Ind.	Proc.
libxslt 1.1.23 PHP	1.0	all	fast	middle	no	no	5	no	no	same
libxslt 1.1.26 PHP	1.0	all	fast	middle	no	yes	4	no	no	same
MSXML 3.0	1.0	all	middle	middle	no	no	4	yes	no	non-proc
MSXML 6.0	1.0	all	middle	middle	no	no	4	yes	yes	proc
Sablotron 1.0.3	1.0	all	middle	small	no	no	5	no	no	non-proc
Saxon 6.5.5	1.0	all	slow	big	yes	yes	2	no	no	non-proc
Saxon HE 9.4.0.2	2.0	all	slow	big	no	yes	2	yes	no	non-proc
XT 20051206	1.0	UTF-8	slow	big	no	yes	11	no	yes	proc
Xalana 2.7.1	1.0	all	slow	big	no	yes	4	no	no	non-proc
xsltproc 1.1.23	1.0	all	fast	small	no	no	5	no	no	same
xsltproc 1.1.26	1.0	all	fast	small	yes	no	4	no	no	same

As expectable, Java processors have the highest time and memory usages. Conversely, command line processors are the fastest ones. Moreover, non-procedural access and non-indented XSLT templates have better time and memory usages than procedural access and indented XSLT templates. Next, all processors support only XSLT 1.0. Of course, exceptions exist and they will be mentioned for individual processors.

Processors with kernel *libxslt* have a problem with namespaces aliases. The advantage is good warnings about using of unsupported elements in XSLT template. Moreover, command line variants of *xsltproc* have better warnings than PHP variants. Version 1.1.26 is slower than 1.1.23 regardless the variant (command line or PHP library). On the other hand, version 1.1.26 is more reliable, version 1.1.23 failed on the test case of category DocBook.

Processors *MSXML 3.0* and *MSXML 6.0* failed on the test case of category DocBook. Processor *MSXML 6.0* had better time usages for procedural access for some cases which is interesting. Moreover, both processors have better time usages for indented XSLT templates, which is interesting too. Version 6.0 is faster than version 3.0 for big input files.

Processor *Sablotron 1.0.3* failed on the test case of category DocBook too. On the other hand, it has good reports of warnings and errors (e.g. report of unsupported used element, unsupported XSLT 2.0 by declaration etc.). A disadvantage is wrong support of namespace aliases and big slowdown with bigger input XML files.

Processors with kernel *Saxon* have the most successfully passed tests. They failed only on 2 test cases. Version 6.5.5 has better time also memory usages than version HE 9.4.0.2. Moreover, version 6.5.5 allows for using of Java functions in XSLT templates as the only one processor from all the tested processors. On the other hand, version HE 9.4.0.2 supports XSLT 2.0 as the only processor from all the tested ones. In addition, version HE 9.4.0.2 is the least affected by bigger input XML files and has better warnings than version 6.5.5.

Processor *XT 20051206* has the most failed tests (total 11). It supports only encoding UTF-8, it does not support namespace aliases and using Java functions in XSLT templates. In addition, it failed on the test of the category Google Search Appliance. It has better time usage for procedural access in some cases, which is interesting.

Last but not least, processor *Xalan 2.7.1* does not support using of Java functions in XSLT templates and namespaces aliases. It is an average XSLT processor.

5 Conclusion

Our aim was to create an XSLT benchmark which is unique after a long time. The resulting tool, called *XSLTMark II*, allows for generating of tests from templates of tests, running tests, generating XML reports, transforming reports into HTML format and testing different types of XSLT processors. In addition, it allows for many extensions. We can add other tests, templates of tests, tested processors

and transformations of reports into other formats. Running of the program can be modified by many parameters. Nevertheless, only few parameters are sufficient for basic running. Thus, its usage is very simple. Possibility of running it on different operating systems is a big advantage too. In addition, it is a command line program, thus it is possible to run it as a component of others scripts. The program is freely available at [9] for possible usage and/or upgrade.

Naturally, we can still identify several possible extensions of *XSLTMark II*. An interesting usage might be to test applications based on XSLT transformations. Thus, it would be similar to, e.g., the *PHPUnit* [8] tool, which is designed for testing of PHP applications. It would be also useful to prepare a driver for conversion of reports, which would generate text summary. And, naturally, extension with other test cases, especially when implemented as a kind of repository, is a possible (continuous) future plan.

References

1. Google, <http://www.google.com>
2. Google Code, <http://code.google.com>
3. MSXML, <http://msdn.microsoft.com/en-en/data/bb190600.aspx>
4. Xalan, <http://xalan.apache.org>
5. XT, Version 20051206 (2005), <http://www.blzn.com/xt/xt-20051206/>
6. libxslt – The XSLT C library for GNOME (2009), <http://xmlsoft.org/XSLT/>
7. RSS 2.0 Specification (March 2009),
<http://www.rssboard.org/rss-specification>
8. PHPUnit (April 2012), <https://github.com/sebastianbergmann/phpunit/>
9. XSLTMark II., version 1.0.0 (2012), <http://xsltbenchmarking.masicek.net/>
10. Keenleyside, J., Barbosa, D., Mendelzon, A.: ToXgene – the ToX XML Data Generator – version 2.3 (February 2005), <http://www.cs.toronto.edu/tox/toxgene/>
11. Dolph, C., Kuznetsov, E.: XSLTMark, XSLT Processor Benchmarks (March 2001), <http://www.xml.com/pub/a/2001/03/28/xsltmark/index.html>
12. Cowan, M., Niksic, H.: Wget – The non-interactive network downloader. GNU Wget version 1.11.4, <http://www.gnu.org/software/wget/>
13. Kay, M.H.: Saxon (December 2011), <http://saxon.sourceforge.net/>
14. Rehm, R., Ohrt, M., Tews, U.: Smarty – template engine, version 3.1.4 (October 2011), <http://www.smarty.net/>
15. Masicek, V.: XSLT Benchmarking (2012),
<http://www.ksi.mff.cuni.cz/~holubova/dp/Masicek.pdf>
16. Mlynkova, I.: XML Benchmarking: Limitations and Opportunities (Technical Report) (2008), <http://www.ksi.mff.cuni.cz/~mlynkova/doc/tr2008-1.pdf>
17. Ghring, P., Hlavnicka, P., Cimprich, P.: Sablotron (February 2010),
http://www.gingerall.com/charlie/ga/xml/p_sab.xml
18. W3C. XSL Transformations (XSLT) Version 1.0 (November 1999),
<http://www.w3.org/TR/xslt>
19. Walsh, N.: The DocBook Schema Version 5.0 (March 2008),
<http://www.docbook.org/specs/docbook-5.0-spec-cd-03.html>

ReMoSSA: Reference Model for Specification of Self-adaptive Service-Oriented-Architecture

Sihem Cherif, Raoudha Ben Djemaa, and Ikram Amous

MIRACL, ISIMS, Cité El Ons, Route de Tunis Km 10,
Sakiet Ezziat 3021, Sfax, Tunisia
sihemcherifs@gmail.com
Raoudha.Bendjemaa@isimsf.rnu.tn
Ikram.Amous@iseecs.rnu.tn

Abstract. Specification of SOA has been used to decrease the complexity of service's development to illustrate the self-adaptive applications. On the one hand, it is a means that provides us the appropriate vocabulary for describing the self-adaptive applications. On the other hand, it grants the key architectural characteristics of self-adaptive service under highly changing environments. In this paper, we present ReMoSSA a formal reference model for specifying self-adaptive Service-Based Applications (SBA). ReMoSSA integrates self-adaptation mechanisms and strategies to provide autonomic and adaptable services. It provides a dynamic monitoring and dynamic adaptation in the design phase. ReMoSSA reduces the cost and the effort of maintenance.

Keywords: Self-adaptation, reference model, ReMoSSA, specification.

1 Introduction

The Web Service allows creating agile and evolutionary SOA, it guarantees the interoperability. Web Service, on the one hand, mitigates the shortcomings of SOA in terms of flexibility. On the other hand, it ensures interoperability with the XML standards. But, the new researches applied on the Web services, which lack sufficient adaptability [7], [8] and flexibility.

Over the last decade, researchers have proposed Self-adaptation in system. It they provide a dynamically adaptation and it resolve the previous limits. Self-adaptation mechanism provides closed loop that adapts the system to changes without human intervention. Self-adaptive Service-Based Applications (Self-adaptive SBA) can be running despite the changes of the dynamic context and the failures in the software and hardware components. But, Service Oriented Architecture does not incorporate mechanisms for self-adaptation; the architecture needs always the human supervision to continue operation under highly changing environments. This human surveillance is an open loop. It remains significantly more challenging than traditional systems [15]. Eventually, we propose to integrate the mechanisms for self-adaptation in the specification phase of the SBA using a reference's model.

Specification has been used to decrease the complexity of service's development; to illustrate the self-adaptive applications. On the one hand, it is a means that provides us the appropriate vocabulary for describing the self-adaptive applications. On the other hand, it grants the key architectural characteristics of self-adaptive service under highly changing environments. In fact, the feedback loop model has been used as a reference to construct a reference model in different application domains, in many cases, such as MAPE-K [10], FORMS [15] and DYNAMICO [14]. However, these researches cannot reveal the coherence and traceability between specifications and implementation. It do not satisfy the major concerns of Self-adaptation as (i) how the system controls the environment (i.e. context awareness); (ii) how the system controls and adapts itself; (ii) and how the system coordinates the monitoring and adaptation in a distributed context [12].

So, we think that, to solve these problems, it is necessary to avoid the human intervention (adaptation in open-loop) and to represent the dynamic context management. As a consequence, we propose, in this article, ReMoSSA a formal reference model for specifying self-adaptive SBA. It integrates self-adaptive mechanisms. ReMoSSA reduces the cost and the effort of maintenance. Our reference model was inspired by FORMS model and the automate element proposed by IBM researchers [10]. ReMoSSA can be used to check if the dynamic monitoring and the dynamic adaptation are being considered in the designs.

The remainder of this article is organized as follows. Section 2 presents a general architecture of context. Section 3 describes our proposed reference model and principal components of ReMoSSA, section 4 describes an application example that we use to explain our reference model and its application. Section 5 describes related works. Finally, Section 6 discusses and concludes the article.

2 Dynamic Context in Self-adaptive Application

The design and implementation of self-adaptive SBA requires a strong knowledge of the system's context and of its evolutions. This dynamic context includes: user context (location, activity, and preferences) [16], application's execution context (network protocol, environment information, and device), platform context (hardware and software resource) [1] and infrastructure context (CPU, OS, RAM capacity).

We have developed our model of context for self-adaptive SBA. It enables the information to be shared among dynamic context and provides a different part of Self-adaptation [1].

A context model defines and stores context data [4]. This model contains the contextual information that is used in the context aware application [2].

Strang and Linnhoff-Popien [13] proposed a context modeling approaches (Key-Value models, Markup scheme models, Ontology based models, Graphical models), which are based on the data structures used for representing contextual information in the system.

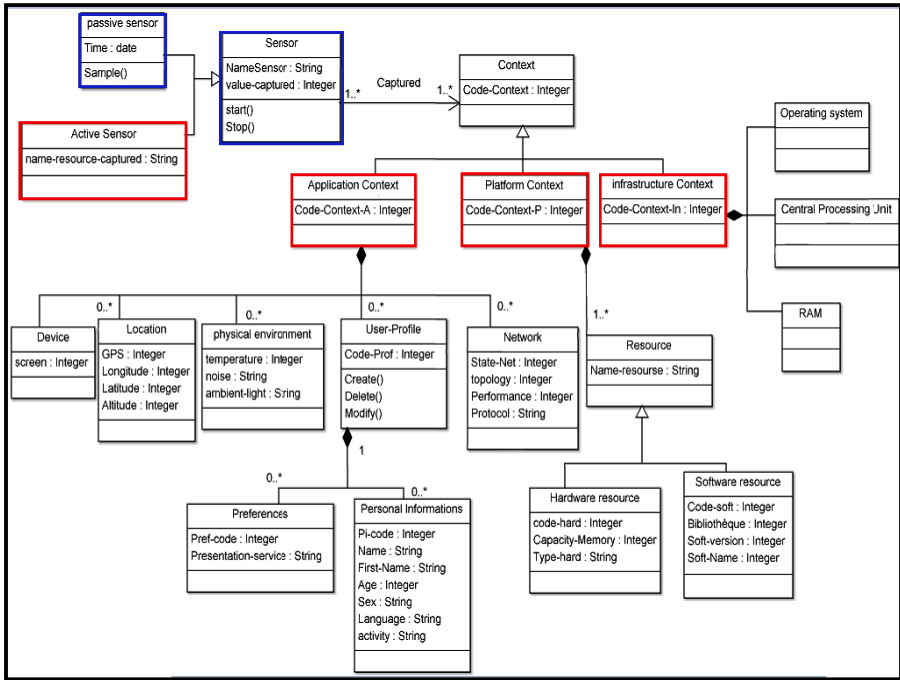


Fig. 1. Context model for self-adaptive SBA

Figure 1 represents the UML diagram which represents the dynamic context. This context contains three elements: application’s context, platform context and infrastructure context. Application context includes the network (topology and performance of the network), the user profile (user preferences, characteristics, activity, etc) [16], the physical environment (temperature, noise) and location (longitude, latitude of terminal).

Service oriented platform is the manner of such application testing, versioning. This platform can range from application aggregation and augmentation context. Platform context contains the hardware resource and software resource where service is running on.

Infrastructure context contains the basic physical structures needed for the system. The infrastructure context includes for example the operating system, central processing unit and RAMS capacity. All context elements are controlled by sensors (passive sensor and active sensor).

The sensors capture three levels: The level of infrastructure, the level of services’ platform and the level of the application.

In our context model, we propose two types of sensors; the active sensor makes observations of the system autonomously and passive sensor makes observations of the system when asked explicitly (the programmers send request for passive sensor). This context model described in this section will be used in our reference model ReMoSSA.

3 ReMoSSA: Our Reference Model

Today, SBA needs a reference to make adaptations according to continuous changes in context. In fact, several domains use the feedback-control loop in order to control autonomously the dynamic services. Our model based on primitives of FORMS model [15], MAPE-K control loop and mechanisms of Self-adaptation for designing a way to divide the different adaptation phases. With our reference model, we aim to contribute to the design of self-adaptive application by making its instances consider these aspects explicitly: (i) the specification of context management as an independent control function to preserve the contextual relevance with respect to dynamic context changes, and (ii) the computational reflection of system.

3.1 Functions of ReMoSSA

ReMoSSA is divided into four main functionalities of the MAPE-K loop, presented in Figure 3: Control, Decision, Plan and Execute [10].

Control function of our model is used to collect dynamic changes of application based on changes context. More precisely the control function is designed to gather relevant events to represent a change of all system components and its environments. This function captures periodically the context to generate events and update the changes. The events that are forwarded to the decision function are events that are different to previous values. In addition, the control functions can periodically captures the application.

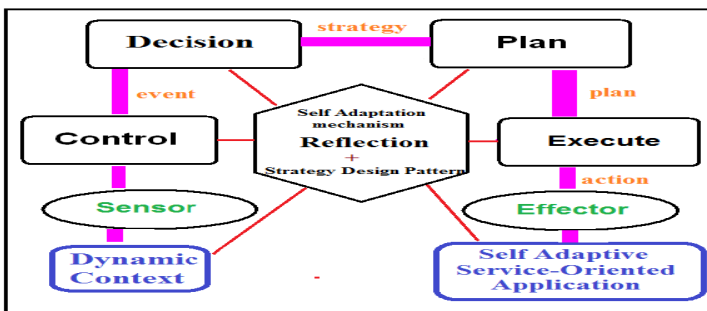


Fig. 2. Different functions of ReMoSSA

The sensors capture the application's execution context, the platform context and the infrastructure context. The platform context contains the running on of the adaptive service. The infrastructure context includes information about the hardware and software resource.

Analysis and decision function exploit the collected events to determine the strategies of adaptation to satisfy system's objectives. When the control function sends an event, the decision function decides if an adaptation is needed then, it chooses an adaptation strategy. **Planning function** puts a set of actions that are used to transform the current system into a new state. A decision result is represented by a

strategy that defines the new state to be achieved, but it does not define how to obtain this state. It is the role of the planning phase. Each type of adaptation refers to a set of actions that can be used for implementation. This phase defines the plans' adaptation based on a set of planning algorithms either local or distributed. **Execute function** uses to implement adaptation actions on the system. The input phase "execute" is a plan that will be used to build new adaptation actions. Execution needs to be efficient due to the dynamism of the context.

Figure 3 shows an overview of ReMoSSA model which extending the primitives of FORMS and adding the new elements required to support dynamic adaptation of SBA.

With our model, we can dynamically add or move conditions, planning strategies, or adaptation actions in order to modify the way the autonomic behavior is implemented in the application.

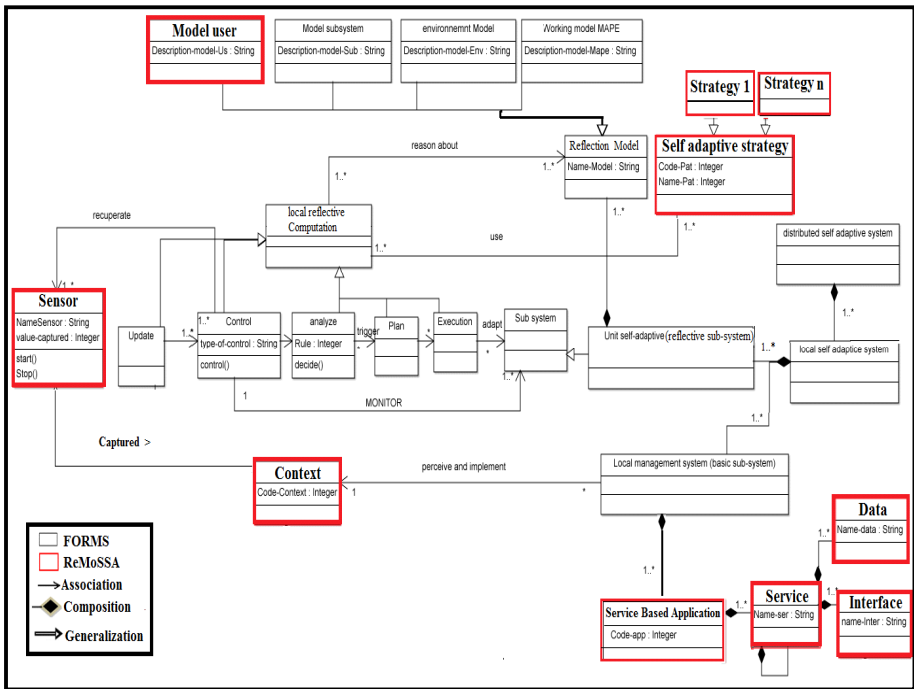


Fig. 3. Our ReMoSSA reference model

After describing the functions of ReMoSSA, we define, in the next section, the self-adaptive mechanisms.

3.2 Self-adaptive Mechanisms in ReMoSSA

ReMoSSA uses self-adaptation mechanisms like reflection mechanisms that can be used to adapt dynamically the behavior of applications. A reflection mechanism

provides the ability of the SBA to observe and to modify its computation [3]. We introduce this mechanism in ReMoSSA for dynamically adapting the service and composing in two levels: the first level is the base level. It is responsible for the description of the computations that the system is supported to do. The second level is the meta-level and defines how the base level computations have to be carried out.

4 Applying the Reference Model

We have applied ReMoSSA to study the case of travel agency. We describe the concepts and elements found within the travel agency via ReMoSSa’s entities.

The purpose of case study is (i) to demonstrate the ability to reason about Self-adaptation mechanisms of the modeled systems, (ii) to control the dynamic change of context. In this section, we study a distributed travel agency application that includes self-adaptive strategies to support adaptability.

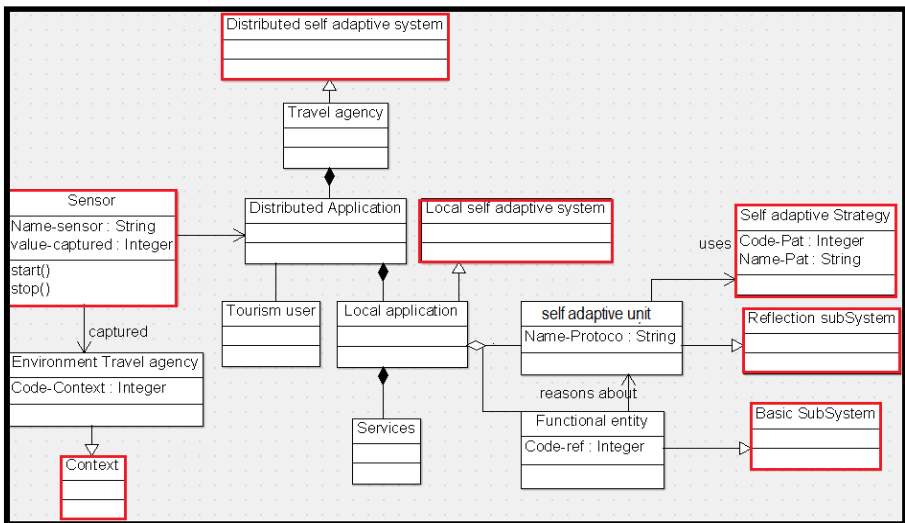


Fig. 4. Travel Agency Case study with ReMoSSA

The figure 4 presents the UML diagram for travel agency using our reference model ReMoSSA. The travel agency system comprise a set of services: booking services, payment service, and research flights service. Many kinds of tourism users want to execute self-adaptive travel agency application in its terminal (mobile phone, PDA and PC).

The service changes dynamically when the context changes. We use sensor components to capture the changes of these contexts.

The Functional entity provides the functionality of services i.e., the functionality to provide booking services. In normal travel agency conditions, each client can have access to application and benefit the travel agency services.

However, the self-adaptive unit uses self-adaptive strategy, when this travel agency detects the problems like hosting server failed.

This problem is analyzed the hosting server and decided to change it without stopping application. In fact, to support robustness to travel agency services, a self-adaptive unit is added to the system for monitoring the execution of application and detecting failures application.

5 Related Works

In this section, we take a look at some research works interested in the possibilities of integrating a reference models in the design of self-adaptive systems. We provide an overview of some of these works.

Kephart and al. [10] propose a feedback loop called MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) which describes the different steps of autonomous systems. But MAPE-K is not formalized.

Weyns and al. [15] present FORMS model (FOrmal Reference Model for Self-adaptation) that uses MAPE-K loop and language Z to formally specify a self-adaptive system. FORMS is composed by a set of modeling primitives that correspond to key points in the self-adaptive software systems, and a set of relationships that govern their composition. However, we learned that in most cases, the primitives must be refined to be really useful for an engineer, but, the primitives FORMS is coarse grain. Second, the reasoning description of a self-adaptive system is specified with the language Z. However, the specification used Z tends to be long. FORMS also do not support coherence and traceability between specifications and implementations. Other researches such as Villegas and al. [14] define reference model called Dynamico (Dynamic Adaptive, Monitoring and Control Objectives model) is a reference model for adaptive software, this model improves engineering self-adaptive systems.

We notice that most of the previous studied works were still insufficient since they do not reveal the coherence and traceability between specifications and implementation. In addition, the context of execution of application; in each model; is not effective. For these reasons, our idea is to provide a generic solution for specifying self-adaptive SBA and in order to solve the existing problems in previous models.

6 Conclusion and Future Works

In this article, we have presented ReMoSSA, a reference model for specifying the self-adaptive service oriented application. This kind of application must deal with highly dynamic contexts and reflection on itself. ReMoSSA aims to incorporate various points of view into a unifying reference model. The strength of our model includes three mechanisms; Reflection, MAPE-K, self-adaptive strategies pattern. They are influenced the majority of existing approaches employed in the construction of self-adaptive applications. As a reference model, ReMoSSA emphasizes the visibility of formal representation. ReMoSSA contains a set of relationships between the entities. It constitutes a guide to design self-adaptive SOA applications.

For future research, there are several possibilities for the extension and validation of our model ReMoSSA: (i) we use ReMoSSA to develop a generic self-adaptable middleware; (ii) we enrich the context model by adding a feedback loop in the capture layer; (iii) we integrate complex algorithms to treat adaptation strategies.

References

1. Ben Djemaa, R., Amous, I., Hamadou, A.B.: Adaptability and adaptivity in the generation of web applications. *JITWE* 4(2), 20–44 (2009)
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing* 2(4), 263–277 (2007)
3. Cazzola, W., Ghoneim, A., Gunter, S.: Reflective Analysis and Design for Adapting Object Run-Time Behavior. In: *OOIS 2002*, Montpellier, France, pp. 242–254 (2002)
4. Chaari, T., Laforest, F., Celentano, A.: Adaptation in Context-Aware Pervasive Information Systems: The SECAS Project. *Int. Journal on Pervasive Computing and Communications* 3(4) (2007)
5. Chen, H.: *An Intelligent Broker Architecture for Pervasive*. University of Maryland, Baltimore County (2004)
6. David, P.-C., Ledoux, T.: WildCAT: a generic framework for context-aware Applications. In: *Proceeding of the 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing, MPAC 2005*, Grenoble, France (November 2005)
7. EL Hog, C., Ben Djemaa, R., Amous, I.: AWS-WSDL: A WSDL Extension to Support Adaptive Web Service. In: *IiWAS 2011*, Ho Chi Minh City, Vietnam (December 2011)
8. EL Hog, C., Ben Djemaa, R., Amous, I.: Towards an UML Based Modeling Language to Design Adaptive Web Services. In: *SWWS 2011*, Monte Carlo Resort, Las Vegas, USA (2011)
9. Erl, T.: *SOA Design Patterns*. The Prentice Hall Service-Oriented Computing Series (January 9, 2009)
10. Kephart, J.O.: The vision of autonomic computing. *IEEE Computer* 36 (2003)
11. Monfort, V., Chérif, S., Chaaban, R.: A Service Based Approach to connect Context Aware Platforms and Adaptable Android for Mobile Users, ch. 13. *IGI Book* (2012)
12. Ruz, C., Baude, F., Sauvan, B.: Using Components to Provide a Flexible Adaptation Loop to Component-based SOA. *IARIA International Journal on Advances in Intelligent Systems*, 32–50 (July 2012)
13. Stang, T., Linnhoff-Popoen, C.: A Context Modeling Survey. In: *Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004-The Sixth International Conference on Ubiquitous Computing*, Nottingham/England (September 2004)
14. Villegas, N.M., Tamura, G., Müller, H.A., Duchien, L., Casallas, R.: DYNAMICO: A reference model for governing control objectives and context relevance in self-adaptive software systems. In: de Lemos, R., Giese, H., Müller, H.A., Shaw, M. (eds.) *Software Engineering for Self-Adaptive Systems*. LNCS, vol. 7475, pp. 265–293. Springer, Heidelberg (2013)
15. Weyns, D., Malek, S., Andersson, J.: FORMS: a FORMAL Reference Model for Self-adaptation. In: *ICAC 2010*, Washington, DC, USA (June 2010)
16. Zghal Rebai, R., Zayani, C.A., Amous, I.: A new technology to adapt the navigation. In: *ICTW 2013*, Rome, Italy (to appear, June 2013)

DSD: A DaaS Service Discovery Method in P2P Environments

Riad Mokadem¹, Franck Morvan¹, Chirine Ghedira Guegan², and Djamel Benslimane³

¹ IRIT, Paul Sabatier University, 118 Rte de Narbonne, 31062, Toulouse, France
{mokadem,morvan}@irit.fr

² IAE, Jean Moulin University, 6 cours Albert Thomas, 69355, Lyon, France
chirine.ghedira-guegan@univ-lyon3.fr

³ LIRIS, Claude Bernard University, 69622, Villeurbanne, France
djamel.benslimane@univ-lyon1.fr

Abstract. Exposing data sources through *Daas* (Data as a Service) services become increasingly important. The *DaaS* service discovery constitutes a real challenge in P2P environments. Although several data source discovery methods take into account the semantic heterogeneity problems by using several domain ontologies (DOs), most of them imposed a topology on the graph formed by DOs and mapping links. In this paper, we propose a *DaaS* Service Discovery (DSD) method without imposing any topology on this graph. Peers, using a common DO, are grouped in a Virtual Organization (VO) and connected in a Distributed Hash Table (DHT). Then, lookups within a same VO consists in a classical search in a DHT. Regarding the inter-VO discovery process, we propose an addressing system, based on the existing mapping links between DOs, to interconnect VOs. Furthermore, a lazy maintenance is adopted in order to reduce the number of messages required to update the system.

Keywords: Large Scale Data Distribution, Data as a Service, Data Source Discovery, Semantic Heterogeneity, Dynamicity, Performance.

1 Introduction

With the constant proliferation of information systems around the globe, the need for decentralized and scalable data sharing and integration mechanisms has become apparent more than ever in a wide range of applications. These applications querying heterogeneous data source spread on a huge number of peers which can join/ leave the system at any moment. Last few years saw new type of services known as *DaaS* (Data-as-a-Service) services [19] where services correspond to calls over the data sources. Besides, users' requirements increase so that their queries often need several sources, thus requiring service composition. The latter consists in combining several *DaaS* services. While initial service composition approaches have been a powerful solution for building value-added services on top of existing ones, the issues of exploiting and managing *DaaS* services in dynamic and large scale P2P environments remain an important challenge. In fact, *DaaS* services are numerous, highly heterogeneous and constantly evolving in such environments. This leads to the

fact that the services discovery process remains a more important issue in a large scale query evaluation. Indeed, the discovery process consists in searching metadata describing *DaaS* services required to process the user query. However, the main obstacle affecting the *DaaS* service discovery mechanism is the semantic heterogeneity between services, e.g., the *DaaS* service associated to the 'Doctor' concept in a medicine field is not identical to the *DaaS* service associated to the 'Doctor' concept in the university area. Indeed, resolving the semantic heterogeneity between the different concepts associated with the *DaaS* services is necessary.

The *DaaS* service discovery approaches can be classified into centralized [2] and decentralized [7]. Central registries, used to store *DaaS* services, are poor at supporting scalability in the Web context when most of the decentralized *DaaS* service discovery methods typically employ flooding and random walk to locate data which results in much network traffic. Dealing with the data source discovery related work taking into account the semantic heterogeneity problems, first works were based on the correspondence between keywords used in the data source schemas [9, 17, 18]. The major drawback of this approach is the maintenance of links in a highly dynamic environment. Other works have adopted the use of a global schema or a global ontology, employed to provide a formal conceptualization of each domain, as a pivot schema [1]. However, designing such ontology remains a complex task in front of the large number of areas in large scale environments. Later, some works proposed the using of different domain ontologies (DOs) [8, 14]. In this paper, we adopt this latter approach which is the most promising because it preserves the autonomy of each DO. Hence, each application domain is associated with one DO. Relationships links called 'mapping links' are established between these DOs in order to define correspondence links between them. In our knowledge, all methods proposed within this approach impose a particular topology on the graph formed by the DOs and mapping links. Imposing a fixed topology as in [15] is a major drawback. Indeed, there are on the Internet available DOs and mapping links between them. The topology of the graph formed by these DOs and mapping links between them is an arbitrary graph. If the topology founded is not suitable for one method, some mapping links must be defined. This is a very hard task. Hence, a good challenge consists in using the existing mapping links without imposing any topology on the graph.

In this paper, a part of the PAIRSE project¹, we extend the data source discovery method proposed in [11] to support the *DaaS* service discovery with considering semantic heterogeneity between concepts associated to these services. The discovery process is particularly important that the user query response is produced from the composition/ filtering of returned *DaaS* services [4]. The proposed *DaaS* Service Discovery (DSD) method is adapted to any mappings link topology. We propose to regroup peers, by expertise domain, in a virtual organization (VO) [12]. In each VO, peers used the same DO as a pivot schema. This allows taking into account the principle of locality [10] that promotes the autonomy of each VO. For reasons of discovery process efficiency, the peers within the same VO are connected in a Distributed Hash Table (DHT) [16]. Thus, the discovery within a single VO consists

¹ This research project is supported by the French National Research Agency under grant number ANR-09-SEG-008, and available at: <https://picoforge.int-evry.fr/cgi-bin/twiki/view/Pairste/Web>

in a classical lookup in a DHT. Concerning the inter-VOs *DaaS service* discovery, the translation of the sought concept between VOs is required in order to propagate the *DaaS* service discovery query. For this aim, we propose an addressing system that permits to interconnect VOs by exploiting the existing mapping links between DOs. Then, a permanent access exists from any VO to other. The proposed method takes also into account the connection/ disconnection of peers into the system (dynamicity property of P2P environments). In order to limit the excessive number of messages exchanged between peers, we adopt a lazy update of the addressing system. This permits a significant maintenance costs reduction especially in the presence of a churn effect [12]. The rest of the paper is organized as follows: Section 2 describes the *DaaS* service composition. Section 3 details the proposed DSD method. Section 4 discusses the system maintenance through our method. A simulation analysis of the proposed method is presented in section 5. The final section contains concluding remarks and future work.

2 DaaS Service Composition

DaaS services provide bridges to access data sources. Nevertheless, while individual *DaaS* services may provide interesting information alone, in a real scenario, users' queries require the invocation of several DaaS by adopting composition approaches.

In this work, we adopt a query rewriting based approach to compose data providing Web services proposed in [3]. Specifically, *DaaS* services are modeled as RDF parameterized Views (RPVs) over DOs. RDF (Resource Description Framework) views capture in a faithful and declarative way the semantic relationships between input and output parameters using ontological concepts and relations whose semantics are well defined in the mediated ontology. RDF views are incorporated within services description files as annotations. Users pose their queries at a given peer on a mediated ontology using SPARQL² query language. Then, the defined RDF views are exploited within WSDL files to discover locally and/or distantly services. Indeed, the peer extracts the different ontological concepts used in the query and launches service discovery requests for services annotated (via their RPVs) with these concepts, firstly in the peer where the query is posed, then the service discovery request is propagated to the others peers. The discovery process is detailed in the next section. The descriptions of discovered services are then sent back to the initial peer, where the relevant services will be selected and composed using an RDF query rewriting algorithm [4, 6]. Finally an execution plan for the composition is generated and executed to provide the user with requested data.

3 DaaS Service Discovery Considering Semantic Heterogeneity

P2P environments are characterized by the presence of a large number of Web services which are highly heterogeneous and constantly evolving. Throughout this section, we present the proposed *DaaS* Service Discovery (DSD) method.

² <http://www.w3.org/TR/rdf-sparql-query/>

3.1 System Architecture and Design

Each application domain accumulates a large number of *DaaS* services. In order to preserve the autonomy of each application domain, we propose to associate each application domain with one DO [14]. Then, relationships links called ‘mapping links’ are established between these DOs in order to define correspondence links between them. We group peers using the same DO in a Virtual Organization (VO). This allows taking into account the principle of locality [10]. Indeed, peers having the same expertise are grouped in the same VO. Let consider a set of DOs which form an undirected graph $G(V, E)$ with V the set of vertices presenting these DOs and E the set of edges presenting the mapping links between these DOs. We note that an edge exists between two vertices v_i and v_j in G if and only if there exists a mapping link between DO_i and DO_j presenting respectively per v_i et v_j with $i \neq j$.

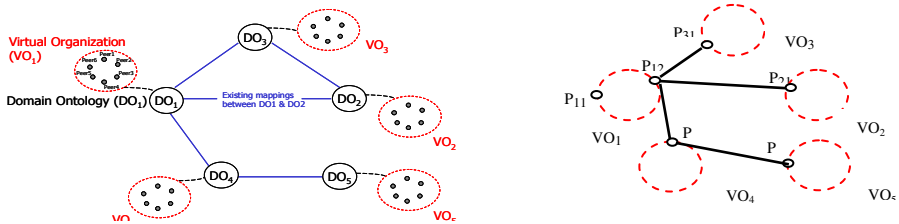


Fig. 1. Example of: a Graph between DOs and Associate VOs (left), Interconnection between VOs through Access Points (right)

For each DO_i in G , we associate a virtual organization VO_i . Each VO regroups a set of peers using the same DO_i as a pivot schema for managing *DaaS services*. We affirm that VO_j is neighbor of VO_i if and only if it exist a mapping link between DO_i and DO_j used respectively per VO_i and VO_j . We notes Neighbor (VO_i) a set of VOs, neighbors of VO_i and connected to VO_i through direct mapping links. Let also $|Neighbor(VO_i)|$ be the number of VOs neighboring of VO_i . To ensure the completeness of discovery results, the graph G must be connected, i.e., there must be a path from any VO to another. In the rest of this paper, we suppose that $G(V, E)$ is connected and its topology is arbitrary. Thus, there is always a path between two vertices in G . This allows a translation between two DOs, e.g., Fig.1- left shows an example of a connected graph of mapping links between DO_1, DO_2, \dots, DO_5 , VO_1 regroups a set of peers using the same domain ontology DO_1 as a pivot schema.

3.2 DaaS Service Discovery Process

Suppose that a user query Q is issued at a given peer. Suppose also that Q could not be resolved with the *DaaS services* founded at a local peer [4]. Then, its evaluation starts with the *DaaS service* discovery step. It consists to discover the metadata describing *DaaS services*, previously published and associated to the concepts present in Q . The *DaaS service* registration step consists in publishing (i) the concerned *DaaS service*, (ii) the associated RDF parameterized View (RPV) which allows to capture

the semantic relationship between the *DaaS service* and the associated concepts over the corresponding DO, (iii) the set of concepts C associated to the *DaaS service*, (iv) the set of properties Pr associated with these concepts on which an additional filtering is applied to the returned *DaaS service*. Using the described configuration of VOs, *DaaS service* discovery queries can be classified into two types: (i) intra-VO *DaaS service* discovery queries and (ii) inter-VO *DaaS service* discovery queries.

Intra-VO *DaaS service* discovery process. The intra-VO *DaaS service* discovery process consists on the *DaaS services* research within a same VO. It does not require any concepts translation since all peers in one VO used the same ontology as a pivot schema. For this aim, we wish to (i) have an efficient mechanism for the *DaaS service* discovery process and (ii) avoid false answers [16]. This means that if the *DaaS service* exists, we want to discover it. For efficient reasons, we have proposed to associate a structured P2P system, e.g., DHT, to each VO. DHTs have proved their efficiency with respect to the scalability and research process. In addition, they have the characteristic to avoid false answers (the case of unstructured P2P systems). The complexity to find a peer responsible of a *DaaS service* is $O(\log(N))$ where N is the number of peers in chord [16], a DHT rooting protocol. The *DaaS service* discovery within a single VO is evaluated according to the routing system of a classic DHT [16]. However, this requires some DHT adjustments in order to index the previously published services. Indeed, we have extended the DHT catalog to support the registration of both the concerned *DaaS service* and the associated set $\{RPV, C, Pr\}$.

Inter-VO *DaaS service* discovery. An inter-VO discovery query providing from a peer ϵVO_i consists to look for metadata describing a *DaaS service* available in VO_j with $i \neq j$. For this aim, we propose an addressing system which assures a permanent access from any VO to other in a dynamic environment in order to permit the translation of each researched concept. The proposed addressing system permits a communication between a peer ϵVO_i and peers in $neighbor(VO_i)$. For each peer $P_i \in VO_i$ we associate $|Neighbor(VO_i)|$ access points. Let AP_i the access point set of P_i . Each access point $P_j \in AP_i$ is one peer of $VO_j \in Neighbor(VO_i)$. Hence, when a peer P_i wants to propagate the discovery query to VO_i , access points and exiting mapping links between DO_i and DO_j are used. In order to avoid that a peer forms a bottleneck or constitutes a single point of failure, we ensure that several access points P_j of a peer $P_i \in VO_i$ reference different peers in VO_j . Fig. 1- right illustrates examples of access points: the bold lines show mapping links between VOs, e.g., $P_{12} \in VO_1$ can communicate with peers of VO_2 thanks to its access point P_{21} .

Every peer receiving a discovery query: (i) execute an intra-VO discovery query and (ii) propagate the discovery query towards neighbors VOs and so on. Suppose that a given peer $P_i \in VO_i$ submits a *DaaS service* query. Hence, a lookup function is evaluated for each $P_j \in AP_i$ in order to search the concept c in $VO_j \in Neighbor(VO_i)$. When P_i contacts its access point, c is translated through the existing mapping rules between DOs. To avoid an endless propagation of a discovery query, we define a Time to Live (TTL) which corresponds to the maximal path length in G than a discovery query can run, i.e., the limit of the query propagation number. The complete inter-VO discovery algorithm is described in [13]. If a *DaaS Service*, at

least, is found, the response is sent to P_i . It contains: (i) metadata of the founded *DaaS* services, (ii) the translation path constituted of a sequence of edges representing the mapping links that the discovery query followed along the discovery process and (iii) the associated RPV describing the semantic relationship between each returned *DaaS* service and c . Finally, a filter is applied to each *DaaS* service through properties ϵPr .

4 System Maintenance

The continuous leaving/ joining of peers is very common in P2P systems. This requires the maintenance of the system. We distinguish two types of maintenance in our system: (i) the DHT maintenance and (ii) the addressing system maintenance that impact the discovery. We will not detail the first case since the system maintenance is done by a classical maintenance of a DHT [16]. Maintaining the addressing system requires the updating of all access points, i.e., defining how the access points are updated. Recall that in P2P Chord systems [16], the connection/ disconnection of one peer generates $\text{Log}^2(N)$ messages when N is the total number of peers in the system.

When a new peer connects to a VO, all its access points must be defined. We based on the same technique used when an access point is not available during the inter-VO discovery. The connection peer algorithm is detailed in [11]. Regarding the peer disconnection process, let's a peer $P_{\text{Disc}} \in VO_i$ disconnects from the system. The first step is to maintain the DHT. This is a classical DHT maintenance [16]. However, the peer P_{Disc} can be an access point for a peer belonging to another VO_j (with $i \neq j$). Hence, the addressing system must be updated. We have adopt a lazy maintenance as in [11]. None of the VO_j ($i \neq j$) is informed by this disconnection. The access points towards this VO_j will be updated during an inter-VO *DaaS* service discovery process. Indeed, during the discovery process, the opportunity is taken to update all access points. This strategy reduces the number of messages required to update the system.

5 Simulation Analysis

We focus on the simulation of a data source discovery process since it is difficult to experiment these peers organized as VOs in a real platform, e.g., Grid'5000³. We based on a virtual simulated network of 10.000 homogenous peers with a single data source by peer. We also used Open Chord [16], one implementation of Chord DHT. In other hand, data sources are exposed as *DaaS* services. Thus, performances of data sources discovery process and *DaaS service* discovery process are almost equivalent.

5.1 Inter-VO Data Sources Discovery

We have compared performance of the DSD method to those of three other data source discovery methods taking into account the semantic heterogeneity problems: (i) data source Discovery according to the Super Peer topology method (DSP) [8], (ii)

³ Grid'5000. www.grid5000.org

data source Discovery method in which the topology is ‘Two by Two peers’ (D2b2) [9] and (iii) data sources Discovery by Flooding method (DFlooding) [5].

When we deal with a single discovery query, D2b2 response times are the largest compared to other three methods. This is due to the longest path traversed to discover data sources within this method. DFlooding and DSD methods show better results in terms of response time. They have almost similar response times with a small advantage to the DFlooding method. However, the graph of mapping links between ontologies in this later must be a complete graph which requires intensive intervention of the administrator. In order to confirm the capability of the proposed DSD method to be scalable, we have varied the number of queries submitted to each peer (between 2 and 500 queries/ sec). Fig.2 shows the ratio of DSD response times over response times of the three compared solutions with a system composed of 100 VO, i.e., with 100 peers/ VO. Experiments show that the DSD response times are significantly reduced compared to D2b2 and DSP methods. When we experiment with 10 queries/ sec, the response times generated by our method are 10 times smaller than those generated by the DSP method. From 10 queries/ sec, the save time is more important. Compared with the DFlooding method, our response times are slowly greater (18%) when we experiment with less than 20 queries/ sec. A save time, whatever small, is obtained from 25 discovery queries/ sec, e.g., a save of 20% with 500 queries/ sec. This is due to the fact that multiple discovery queries in DSD may require the intervention of several different access points when these queries generate some bottleneck at some peers in the DFlooding method. Hence, it seems more reasonable to have more than 20 queries/ sec in a large scale environment. Furthermore, with DSD we can use DOs and mapping links available on Internet and add some mapping links if the graph founded is not connected. In the DFlooding method, a more important number of mapping links must be defined to have a complete graph.

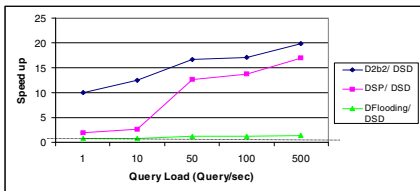


Fig. 2. Speed up (Response Times)

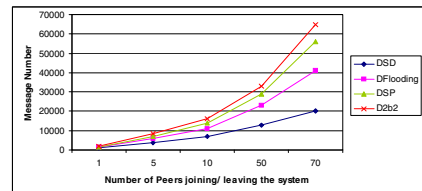


Fig. 3. Impact of the Connected/ Disconnected Peers on the System Maintenance

5.2 System Maintenance: Simulation Analysis

We evaluate the maintenance costs by measuring the number of messages generated to maintain the system when peers join/leave the system but the total number of peers stays appreciatively constant (a system composed of 10 VO with 1000 peers/ VO). It is clear that maintaining a DHT generates greatest costs especially when several peers join/leave the system. But, this is valuable for all the compared solutions. Fig. 3

shows the number of the required messages to maintain the system in the four compared solutions. The number of messages needed to maintain the system with the DSP and D2b2 methods is the higher. Indeed, if a super peer leaves or arrives in the system, all the ‘leaves’ peers should be updated by using the DSP method. The D2b2 method generates the most important maintenance cost. This is due to the topology used. When 10 peers join/ leave the system, better results are observed for the DSD method which requires less than 7200 when the DFlooding method requires 11000 messages to maintain the system. In fact, most of messages in the DSD method are essentially those required to update the DHT. The use of a lazy maintenance allows significant reduction in the number of these messages needed to update access points. Thus, access points of a peer referencing peers which have leaving the system are updated only when the discovery process occurs in the DSD method when all peers must be contacted to update their access points in the DFlooding method.

6 Conclusion and Future Work

The proposed *DaaS* Service Discovery (DSD) method takes into account the semantic heterogeneity problems in P2P environments. For this aim, we group all peers using the same domain ontology (DO) in a virtual organization (VO). Within a VO, the *DaaS* service discovery process is based on a classical lookup in a DHT (intra-VO discovery). Regarding the inter-VO discovery process, we have proposed an addressing system based on the exiting mappings between various DOs without imposing any topology on the graph formed by these DOs and mapping links. Our discovery method allows a permanent access between virtual organizations in a dynamic environment. Furthermore, we adopt a lazy maintenance in order to decrease the update cost generated by the continuous joining/ leaving of peers to the system.

The Simulation analysis showed a significant improvement of response times for the inter-VO discovery queries especially with an important number of simultaneous discovery queries. It shows also a significantly reduction of the maintenance cost generated by the frequent joining and leaving of peers. Further work includes more performance studies especially with a high number of peers in a real platform.

References

1. Alking, R., Hamerlain, A., Morvan, F.: Ontology-Based Data Source Localization in a Structured P2P Environment. In: Proc. of Int. Symposium IDEAS, Coimbra, Portugal, pp. 9–18 (2008)
2. Atkinson, C., Bostan, P., Hummel, O., Stoll, D.: A practical Approach to Web Service Discovery and retrieval. In: Proc. of Int. Conf. ICWS 2007 (2007)
3. Barhamgi, M., Benslimane, D., Medjahed, B.: A Query Rewriting Approach for Web Service Composition. *IEEE Transactions on Services Computing (TSC)* 3(3), 206–222 (2010)
4. Barhamgi, M., Ghedira, C., Benslimane, D., Tbahriti, S.E., Mrissa, M.: Optimizing DaaS Web Service Based Data Mashups. In: Proc. of the Int. IEEE Conf. SCC, pp. 464–471 (2011)

5. Chawathe, Y., Ratnasamy, S., Breslau, L.: Marking Gnutella-like P2P Systems Scalable. In: Proc. of the Int. Conf. SIGCOMM 2003, Karlsruhe, Germany, pp. 407–418 (2003)
6. Chen, H.: Rewriting Queries Using View for RDF/RDFS-Based Relational Data Integration. In: Chakraborty, G. (ed.) ICDCIT 2005. LNCS, vol. 3816, pp. 243–254. Springer, Heidelberg (2005)
7. Comito, C., Mastroianni, C., Talia, D.: A Semantic-aware Information System for Multi-Domain Applications over Service Grids. In: IPDPS, Rome, Italy (2009)
8. Faye, D., Nachouki, G., Valduriez, P.: Semantic Query Routing in SenPeer, a P2P data Management System. In: Proc. of the Int. Conf. on Network Based System Information Systems, NBIS, Germany (2007)
9. Halevy, A.Y., Ives, Z.G., Mork, P., Tatarinov, I.: Piazza: Data Management Infrastructure for Semantic Web Applications. In: Int. Conf. WWW, Budapest, Hungary (2003)
10. Harvey, N., Dunagan, M.B.J., Jones, M.B., Saroiu, S., Theimer, M., Wolman, A.: Skipnet: A Scalable Overlay Network with Practical Locality Properties. In: Proc. of USITIS, Seattle (2003)
11. Ketata, I., Mokadem, R., Morvan, F.: Resource Discovery Considering Semantic Properties in Data Grid Environments. In: Hameurlain, A., Tjoa, A.M. (eds.) Globe 2011. LNCS, vol. 6864, pp. 61–72. Springer, Heidelberg (2011)
12. Mokadem, R., Hameurlain, A., Tjoa, A.M.: Resource Discovery Service while Minimizing Maintenance Overhead in Hierarchical DHT Systems. In: Proc. Int. Conf. iWAS 2010, Paris, pp. 628–636. ACM (2010)
13. Mokadem, R., Morvan, F., Ghedira Gueguan, C., Benslimane, D.: DSD: A DaaS Service Discovery in P2P Environments. IRIT research report, RR-2013-31-FR (June 2013)
14. Navas, I., Sanz, I., Aldana, J.F., Berlanga, R.: Automatic Generation of Semantic Fields for Resource Discovery in the Semantic Web. In: Andersen, K.V., Debenham, J., Wagner, R. (eds.) DEXA 2005. LNCS, vol. 3588, pp. 706–715. Springer, Heidelberg (2005)
15. Nejdil, I., Wolf, W., Qu, B.: Edutella: AP2P networking infrastructure based on RDF. In: 11th Int. World Wide Web Conf., Hawaii, USA, pp. 604–615 (May 2002)
16. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnany, H.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In: Proc. of ACM Conf. SIGCOMM, California, USA (2001)
17. Sartiani, C., Manghi, P., Ghelli, G., Conforti, G.: XPeer: A Self-organizing XML P2P Database System. In: Proc. of Int. EDBT Workshop on P2P and Databases, Heraklion, Greece (2004)
18. Ng, S.W., Chin Ooi, B., Tan, K., Zhou, A.: PeerDB: A P2P-based System for Distributed Data Sharing. In: Proc. of Int. Conf. of Data Engineering, Bangalore, pp. 633–644 (2005)
19. Vaculin, R., Chen, H., Neruda, R., Sycara, K.: Modeling and Discovery of Data Providing Services. In: Proc. of Int. Conf. on Web Services, pp. 54–61. IEEE Computer Society, Washington, DC (2008)

Part II

**Special Session on
Big Data: New Trends and
Applications**

Designing Parallel Relational Data Warehouses: A Global, Comprehensive Approach

Soumia Benkrid^{1,2}, Ladjel Bellatreche¹, and Alfredo Cuzzocrea³

¹ LIAS/ISAE-ENSMA, France

soumia.benkrid,bellatreche@ensma.fr

² National High School for Computer Science (ESI), Algeria

s_benkrid@esi.dz

³ ICAR-CNR and University of Calabria, Italy

cuzzocrea@si.deis.unical.it

Abstract. The process of designing a parallel data warehouse has two main steps: (1) fragmentation and (2) allocation of so-generated fragments at various nodes. Usually, we split the data warehouse horizontally, allocate fragments over nodes, and finally balance the load over the nodes of the parallel machine. The main drawback of such design approach is that the high communication cost. Therefore, *Data Replication* (DR) has become a requirement for availability on the one hand but also for minimizing the communication cost on the other hand. In this paper, we present a *redundant allocation algorithm for designing shared-nothing parallel relational data warehouses*, which is based on the well-known fuzzy *k-means clustering algorithm*.

1 Introduction

Today volumes of data are increasing more and more due to the rise of new infrastructures, such as *Clouds* [1], and new devices, such as sensors [11]. On the other hand, social networks (e.g., Facebook, Twitter and LinkedIn) collect billions of data bytes, and predicting the behavior of users in order to improve their services via analyzing so-collected large data volumes is becoming increasingly hard. As a consequence, traditional Data Warehouses (DW) have become obsolete and Parallel Relational Data Warehouses (PRDW), instead, have been proposed as a robust and scalable platform for storing, processing and analyzing large volumes of data within the layers of modern analytics infrastructures. Similarly, a large number of software companies are positioned around the market with the goal of providing business intelligence solutions on top of large volumes of data, such as Teradata¹, Netezza², and so forth. In line with these major trends, Small and Medium-sized Enterprises (SME) are defining new classes of jobs dealing with so-called *Big Data Actors* such as Data Architect, Data Visualizer, Data Analyst etc, thus exposing a clear commercial demand. This despite

¹ <http://www.teradata.com/>

² www.ibm.com/software/fr/data/netezza/

Big Data software platforms still remain costly for PME in terms of license fees and costs of installation and maintenance (stirred-up by the actual economic crisis).

Under a general view, designing a PRDW comprises four main steps: (1) choosing the hardware architecture; (2) partitioning the target DW; (3) allocating the so-generated fragments over available nodes; (4) defining efficiency query processing strategies. Currently, several types of hardware architecture are available, such as *Shared-Nothing*, *Shared-Disk*, *massively parallel processors* and *Clusters of workstations*. Shared-Nothing architecture has been proposed by DeWitt [12] as the reference architecture for supporting high-performance data warehouses modeled in terms of relational star schemas. As the choice of the hardware architecture is influenced by price, high-performance features, extensibility and data availability [8], Clusters of workstations are very often used as a valid alternative to Shared-Nothing architectures (e.g., [5]).

According to this low-cost technology solution, the target DW is divided into disjoint units called *partitions* that do not introduce any loss or addition of information with respect to the corresponding combination of partitions kept in the original DW. Data partitioning can be done horizontally or vertically, alternatively. Horizontal partitioning is essentially used to design PRDW. Data allocation consists in placing generated fragments over nodes of a reference parallel machine. This allocation may be either *redundant* (with replication) or *non redundant* (without replication). Once fragments are placed, global queries are executed over the processing nodes according to parallel computing paradigms. Load balancing is usually performed by means of the *multi-reordering* process according to which multiple processors that have small average loads are selected in order to participate to the load balancing. According to this schema, each free processor is moved as to becoming adjacent (according to the node network topology) to a high-loaded processor, the load of which is then shared with the (newly-introduced) free processor. This so-determined *data migration* task may cause high communication costs, which overall lower the global throughput of the PRDW architecture. From active literature (e.g., [2]), it is well-understood that communication cost is a factor that must be mastered depending on the available infrastructure, and that most of data access must be local (for efficiency purposes). Therefore, *data replication* has become a strict requirement in PRDW architectures in order to guarantee avoiding bottlenecks and reducing communication costs. To this end, replication aims at (a) ensuring data availability and fault tolerance, (b) improving data locality by following the criterion of placing a job at the same node where its data are located, and (c) achieving load balancing by distributing work across data replicas.

On the basis of the guidelines above, here we assert that PRDW design can be modeled as the following tuple: $\langle DP, DA, DR, LB, QP \rangle$, where DP represents the data partitioning schema, DA the data allocation schema, DR the data replication schema, LB the load balancing scheme, and QP the parallel query processing model, respectively. Unfortunately, each one of the sub-tended problem of the main PRDW design problem is NP-hard [2,4,20].

1.1 Contributions of This Research

Under a broader vision, the PRDW design problem can be thought as a *set of services* offered by *actors* which communicate and cooperate among them in order to obtain a high throughput in the whole PRDW architecture. In our research, we particularly introduce five actors: *Partitioner*, *Allocator*, *Replicator*, *LoadBalancer*, *ParallelQueryProcessor*, each one focusing on a particular PRDW design aspect. By inspecting the active literature, comprehensive surveys of state-of-the-art research on PRDW design issues exist, but still researchers focus the attention on PRDW issues in an isolated manner. In fact, some focus on the data partitioning problem (e.g., [22,21,19]), other on the data allocation problem (e.g., [4,2,18]), or the data replication problem (e.g., [10,13,17]), or the parallel query processing problem (e.g., [3,15,16]). As a consequence, two main limitations may penalize the PRDW design phase: (1) neglecting the inter-dependency among the different-but-related PRDW design issues and (2) adopting heterogeneous metrics in order to identify the quality of the final solution (indeed, each one of the five actor considers a different metric to this end).

In this paper, we propose a novel method for designing PRDW over parallel machines. The basic idea is to *consider the interaction among the different aspects of the main PRDW design problem in order to use a unique cost model that smoothly unifies all phases*. Packaging the PRDW design issues as a unified process that cements PRDW design phases and increases the omniscience of the actors. Since data partitioning plays an important role in the whole PRDW design, we consider it as the first important step in this design. During the fragmentation phase, the Partitioner should consider that the target RDW need be partitioned as to make it "good" for the Allocator, Replicator, LoadBalancer and ParallelQueryProcessor. The design quality is finally measured by the unified cost model. In other words, each potential fragmentation solution is tested for allocation, replication, and load balancing, respectively. The solution having the minimum cost is finally selected for the PRDW design of the EDRP

2 Fragment-Driven PRDW Design Problem Formulation

In our approach, the fragmentation process is the core of the PRDW design methodology, the quality of PRDW design methodology itself strongly depends on the quality of the fragmentation process. We name this methodology as fragment-driven PRDW design methodology. The fragment-driven PRDW design problem can be formalized as a *Constraint Optimization Problem*. Consider the following items:

- A Relational Data Warehouse \mathcal{RDW} modeled using a star schema composed of one fact table \mathcal{F} and d di-mension tables $\mathcal{D} = \{D_1, D_2, \dots, D_d\}$ - as in [16], we suppose that all dimension tables are replicated over the nodes of the database cluster and are in their main memory;
- A database cluster machine \mathcal{DBC} with \mathcal{M} nodes $\mathcal{N} = \{N_1, N_2, \dots, N_M\}$;

- A set of star join queries $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_L\}$ to be executed over DBC , being each query Q_l characterized by an access frequency f_l ;
- the *maintenance constraint* \mathcal{W} , such that $\mathcal{W} \succ \mathcal{N}$, representing the number of fragments \mathcal{W} that the designer considers relevant for his/her target allocation process, this number must be larger than the number of nodes ($\mathcal{W} \succ \mathcal{M}$);
- The *replication constraint* \mathcal{R} , such that $\mathcal{R} \leq \mathcal{M}$, representing the number of fragment copies that the designer considers relevant for his/her parallel query processing;
- The *attribute skewness constraint* θ , representing the degree of non-uniform distribution of the values of the sub domain of an attribute admitted by the designer for the selection of the attributes of fragmentation;
- The *data placement constraint* α representing the degree of data placement skew that the DWA admit for the placement of data;
- the *Load Balancing constraint* δ representing the data processing skew that the designer considers relevant for his/her target query processing;

The problem of designing a PRDW from DWS over the database cluster DBC consists in *fragmenting the fact table \mathcal{F} into \mathcal{NF} fragments and allocating them and the replicated fragments over different DBC nodes such that the total cost of executing all the queries in \mathcal{Q} can be minimized while all constraints modeling the problem satisfied.*

3 The $\mathcal{F}\&\mathcal{A}\&\mathcal{R}$ Approach

In this Section, we describe in detail our proposed PRDW design methodology, which we name as $\mathcal{F}\&\mathcal{A}\&\mathcal{R}$, following our previous proposal [7,6]. To select horizontal partitioning schema, we adapt our genetic algorithm proposed in [5]. Representing chromosomes that model candidate fragmentation schema is the most probing tasks when applying genetic algorithms to the PRDW design problem. Each chromosome may be represented as a *multidimensional array* that models the partitioning domain of a fragmentation attribute. To identify the partitioning attribute candidate, we perform the following tasks: (1) Extracting of all selection predicates exploited by the input queries. (2) Assigning to each dimension table D_i ($1 \leq i \leq n$) the set of selection predicates they are involved to, denoted by $SSPD_i$. (3) Ignoring dimension tables D_i having an empty set $SSPD_i$ (i.e., they will not participate in the fact table fragmentation process). (4) Identifying the set fragmentation attribute candidates. (5) Eliminating attributes having high skew and that do not satisfy the attribute skew constraint. (6) Decomposing domain values of each fragmentation attribute into sub-domains (each sub-domain may be represented by a simple predicate along with its selectivity factor defined on the fact table).

Once the set of fragmentation attribute is identified, our proposed genetic algorithm generates a random population that contains several chromosomes. For each chromosome, our algorithm checks if it satisfies the maintenance constraint

($NFF_i \leq W$), where NFF_i represents the number of fragments. If it is the case, this chromosome is kept in the population; otherwise, merges operations are applied to reduce its number of fragments. Once initiation population created, our genetic algorithm performs operators such as crossover and mutation to improve the quality of this population. The application of these operators is monitored by an evaluation function, which allocates the generated fragments of each valid chromosome over the nodes of the parallel machine. Once this allocation has done the cost of executing queries over nodes is calculated. At the end of this algorithm, the chromosome that offers the minimum cost represents the fragmentation schema. In the next section, we are describing how the mutation operator is done.

The data allocation problem consists to determine the best placement of a set of fragments over database cluster nodes to minimize the cost of answering a workload \mathcal{Q} . The problem of data allocation in distributed and parallel databases (and data warehouses as well) can be formalized as a *clustering problem*. In fact, the clustering problem involves in placing a set of entities into a given number of groups according to a given measure of their tendency to be used together. This turns to be involved in answering a given set of queries.

The fragment allocation is closely related to the fragment replication problem. In other words, the data allocation algorithm is in charge of deciding whether fragments will be replicated or not. To this end, we propose using a *fuzzy clustering method*, namely the *fuzzy k-means clustering algorithm* [9]. In fuzzy clustering techniques, data points can belong to more than one cluster, and associated with each of the points are so-called "membership grades" which show the degree at which data points belong to the different clusters. The fuzzy clustering is often better suited than classical clustering techniques as there is often no sharp boundaries among clusters of data. In fuzzy clustering, membership degrees between 0 and 1 are used instead of crisp assignments of data in clusters. The underlying principle in fuzzy clustering is assigning data elements to multiple clusters, with varying degree of membership.

Allocating the so-generated fragments of each chromosome, we propose a new allocation procedure based on fuzzy clustering of fragments. Let us formulate the fragment allocation problem as follows.

Consider a set of fragments $\mathcal{F} = \{F_1, F_2, \dots, F_{NF}\}$ with dimension in the Euclidean space R^d , i.e., $F_j \in R^d$. The problem of fragment allocation via fuzzy clustering consists in performing a partitioning of these fragments into M fuzzy sets with respect to a given criterion, being M the number of *DBC* nodes. The criterion is usually determined as to optimize an *objective function*. The result of the fuzzy clustering can be expressed by a partitioning matrix U such that $U = [i][j] = u_{ij}$, such that $i = 0..M - 1$ and $j = 1..NF$, where u_{ij} is a value in $\{0, 1\}$, which expresses the membership degree. Besides this, there exists the constraint on u_{ij} stating that the total membership values of fragments $F_j \in \mathcal{F}$, with $j = 1..NF$, in all classes is equal to 1, i.e.:

$$\sum_{i=0}^{M-1} u_{ij} = R \tag{1}$$

The objective function f_O to be minimized is defined as follows:

$$f_O = \sum_{k=0}^{NF-1} \sum_{i=0}^{M-1} u_{ij}^m \|X_k - V_i\|^2 \tag{2}$$

wherein: (i) $m > 1$ is a degree of fuzziness that governs the influence of membership degrees, (ii) X_k is the vector of data points, (iii) V_i is the center of cluster C_i , (iv) $\|X_k - V_i\|^2$ represents the Euclidean distance between X_k and V_i . The steps of our proposed allocation procedure are the following:

1. **Construction of the Fragment Usage Matrix (FUM):** FUM models the usage of fragments according to the set of queries in \mathcal{Q} . FUM contains queries as rows and fragments as columns. The value $FUM[i][j]$, such that $1 \leq i \leq L$ and $1 \leq j \leq NFF$, is equal to 1 if the query Q_i involves the fact fragment F_j ; otherwise, it is equal to 0. An additional column is added to represent the *access frequency* f of each query.

Example 1: Let $F = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8\}$ and $Q = \{Q_1, Q_2, Q_3, Q_4\}$ be the set of so-generated fragments and queries, respectively. A possible FUM of the running example is shown in Table 1.

Table 1. FUM of the running example

Queries	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	f
Q_1	1	1	1	0	1	0	1	0	20
Q_2	1	1	1	1	0	0	0	0	35
Q_3	0	0	1	0	1	1	1	1	30
Q_4	1	1	1	1	1	1	1	1	15

2. **Representation of Each Fragment in R^2 :** each fragment F_i is represented in the two-dimensional space R^2 by coordinates (x, y) . These coordinates of a fragment F_i in R^2 are based on the frequency of queries that do not involve the fragment F_i .

Example 2: The fragment representation associated to the FUM of Table 1 is depicted in Figure 1.

3. **Construction of the Fragment Membership Matrix (FMM):** FMM models the membership degree of each fragment F_k with respect to the cluster C_i according to the set of queries in \mathcal{Q} . FMM contains fragments as columns and clusters as rows. The value $FMM[j][i]$, such that $0 \leq i \leq NFF - 1$ and $0 \leq j \leq M - 1$, is a value in $[0, 1]$ modeling the membership degree of F_k to C_i , given by the *fuzzy k-means clustering algorithm* [9].

Example 3: Based on the fragment representation of Figure 1, the associated FMM of the running example is shown in Table 2.

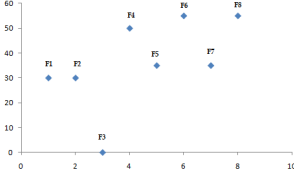


Fig. 1. Fragment representation associated to the FUM of Table 1

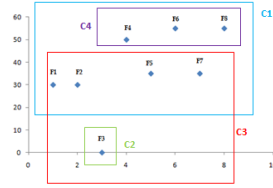


Fig. 2. Fragment clustering associated to the FMM of Table 2

Table 2. FMM of the running example

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
C_1	5,01E-03	5,92E-03	3,60E-09	6,66E-02	9,70E-01	5,28E-03	9,79E-01	1,37E-02
C_2	2,79E-04	2,72E-04	1,00E+00	6,07E-03	7,97E-04	6,94E-04	7,83E-04	1,81E-03
C_3	9,94E-01	9,93E-01	4,87E-09	3,75E-02	2,64E-02	3,27E-03	1,77E-02	8,28E-03
C_4	4,35E-04	4,29E-04	0,00E+00	8,90E-01	2,84E-03	9,91E-01	2,82E-03	9,76E-01

4. **Fragment Clustering:** to generate groups of fragments into clusters, we make use of the basic principle that *larger membership values indicate higher confidence in the assignment of objects to the actual cluster*. As a consequence, on this main insight, we sort membership values in descending order and we assign the fragment F_k to the \mathcal{R} first clusters, being \mathcal{R} the replication degree, such as the data placement constraint is satisfied. At the end of this step, a set of clusters $C = C_0, \dots, C_{M-1}$ is generated, such that each one represents a sub-set of fragments.

Example 4: The fragment clustering associated to the FMM of Table 2 is depicted in Figure 2.

5. **Construction of Fragment Placement Matrix (FPM):** FPM models the positions of a fragment across nodes (recall that fragment replicas may exist). To this end, FPM rows model fragments, whereas FPM columns model nodes. $FPM[i][m] = 1$, with $1 \leq i \leq NF$ and $1 \leq m \leq M$, if the fragment F_i is allocated on the node N_m in \mathcal{N} , otherwise $FPM[i][m] = 0$. Our allocation procedure considers clusters as "movable units" during allocation. Clusters are placed in round robin fashion over the nodes.

Example 5: The generated clusters of Figure 2 are placed in round robin over processing nodes; the associated FPM is shown in Table 3.

4 The $\mathcal{F}\&\mathcal{A}\&\mathcal{R}$ Query Processing Framework

Once the fragmentation schema is generated and the so-generated fragments are placed, *global queries* posed to the data warehouse are then re-written over fragments and evaluated on the database cluster DBC . The ideal parallel query processing method optimizes a smaller set of queries and tries to minimize the

Table 3. FPM of the running example

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
N_1	1	1	0	1	1	1	1	1
N_2	0	0	1	0	0	0	0	0
N_3	1	1	1	0	1	0	1	0
N_4	0	0	0	1	0	1	0	1

total execution cost for the entire set of queries. To evaluate a given query, we should first identify its valid fragments and their localizations across nodes. Since our allocation is redundant (i.e., each fragment can have several placements by means of replicas), we should use a *scheduler* to find the best allocation of each sub-queries. It should be noted that each valid fragment will give rise to a sub-query. The query processing of $\mathcal{F}\&\mathcal{A}\&\mathcal{R}$ can be formalized as follows. Given:

- a set of fragments $\mathcal{F} = \{F_1, F_2, \dots, F_{NF}\}$, being each fragment F_i , with $1 \leq i \leq NF$, characterized by its size $Size(F_i)$;
- a database cluster machine *DBC* having \mathcal{M} nodes $\mathcal{N} = \{N_1, N_2, \dots, N_M\}$;
- a set of star queries $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_L\}$ to be executed over *DBC*, being each query Q_l , with $0 \leq l \leq L - 1$, characterized by an access frequency f_l ;
- the *processing skew constraint* δ representing the data processing skew that the designer considers relevant for his/her target query allocation process;

determine the following state function:

$$isAllocated(Q_i, N_j) = \begin{cases} 1 & Q_i \text{ on } N_j \\ 0 & \text{otherwise} \end{cases}$$

by minimizing the total query processing cost due to evaluating queries in \mathcal{Q} while maximizing the productivity of each node in \mathcal{N} , subject to a fixed processing skew constraint δ that represents the data processing skew that the designer considers relevant for his/her target allocation process, the data placement skew factor.

The above-introduced query processing framework defines a NP-hard problem, which is similar to a *Dual Bin Packing Problem (DBPP)* [14]. To provide sub-optimal solutions to this problem, we propose a proper *greedy algorithm* that is in charge of executing the query scheduling for supporting star query evaluation against the parallel machine (see Algorithm 1).

Focus the attention on Algorithm 1. First, we identify the valid fragments and their associated sub-queries, the number of valid fragments and the set of valid node need for the execution of the sub-queries (lines 1 – 4). Next, we estimate the processing time PTS_{Q_i} needed to evaluate the query Q_j (lines 5) and we initialize the processing bound *MPS* (line 6). We then sort valid fragments in descending order (line 7) and, for each so-generated sub-query (line 8), we perform the following steps: (1) select the valid nodes; (2) calculate load of each valid node; (3) pick the sub-query the node having the largest residual capacity (lines 9 – 12). This finally realizes the scheduling of sub-queries on fragments and their replicas, so that giving the support for their evaluation.

Algorithm 1. Query Allocation (M node, Q_j Query)

- 1: Let **ListFrag** the list of valid fragments for Q_j .
- 2: Let **NumberFrag** the number of fragments in **ListFrag**;
- 3: Let **NumberValideNode** the number of valid nodes for the fragments in **ListFrag**;
- 4: Let **ListSubQuery** the sub-query list : */**each valid fragment will give rise to a sub-query**/*
- 5: Estimate $SizeQ$ the number of Inputs/Output (IO) needed to execute Q_j ;
- 6: Calculate MPS be mean data processing of Q_j ;

$$LB = \frac{1}{\sum_{j=1}^{NumberValideNode} \frac{1}{j^{\delta}}} \times SizeQ \quad (3)$$

- 7: Sort **ListFrag** according to their size in descending order;
 - 8: **for** $i = 1$ to $NumberFrag$ **do**
 - 9: Get the valid nodes for the i^{th} fragment in **ListFrag** and store them in the list **ListNode**;
 - 10: Calculate the load of each node from **ListNode**;
 - 11: Assign F_i to the node with largest residual capacity;
 - 12: **end for**
-

Once the query allocation process has done we calculate the executing cost of the given workload \mathcal{Q} over the M nodes of the \mathcal{DBC} in terms of number of inputs outputs (IOs). It is given by the following equation:

$$\sum_{i=1}^L MAX_{1 \leq j \leq M} \left(\sum_{i=1}^{NF} MUF[i][k] \times MPF[i][j] \times Taille(F_i) \right) \quad (4)$$

5 Conclusions and Future Work

In this paper, we showed the interest to consider the PRDW problem as an unified problem. We have presented a novel design approach called $\mathcal{F\&A\&R}$, which follows our previous proposal in [5,6]. An original Redundant data allocation based on fuzzy logic is integrated into $\mathcal{F\&A\&R}$. Our cost model which evaluates the quality of our solution integrates the concepts of all phases. To reduce the complexity of our solution, we considered low cost execution algorithms. Future work is focused on : (1) development of advanced algorithms that parallelize the various steps of PRDW design, and (2) extending our cost model by considering the interaction among queries.

References

1. Agrawal, D., Das, S., El Abbadi, A.: Data Management in the Cloud: Challenges and Opportunities. Synthesis Lectures on Data Management. Morgan & Claypool Publishers (2012)
2. Ahmad, I., Karlapalem, K., Ghafoor, R.A.: Evolutionary algorithms for allocating data in distributed database systems. In: Distributed Database Systems, Distributed and Parallel Databases, pp. 5–32 (2002)

3. Akal, F., Böhm, K., Schek, H.-J.: OLAP query evaluation in a database cluster: A performance study on intra-query parallelism. In: Manolopoulos, Y., Návrat, P. (eds.) ADBIS 2002. LNCS, vol. 2435, pp. 218–231. Springer, Heidelberg (2002)
4. Apers, P.M.G.: Data allocation in distributed database systems. *ACM Transactions on Database Systems* 13(3), 263–304 (1988)
5. Bellatreche, L., Benkrid, S.: A joint design approach of partitioning and allocation in parallel data warehouses. In: Pedersen, T.B., Mohania, M.K., Tjoa, A.M. (eds.) DaWaK 2009. LNCS, vol. 5691, pp. 99–110. Springer, Heidelberg (2009)
6. Bellatreche, L., Benkrid, S., Crolotte, A., Cuzzocrea, A., Ghazal, A.: The *f&a* methodology and its experimental validation on a real-life parallel processing database system. In: CISIS 2012, pp. 114–121 (2012)
7. Bellatreche, L., Cuzzocrea, A., Benkrid, S.: *F&A*: A methodology for effectively and efficiently designing parallel relational data warehouses on heterogeneous database clusters. In: Bach Pedersen, T., Mohania, M.K., Tjoa, A.M. (eds.) DAWAK 2010. LNCS, vol. 6263, pp. 89–104. Springer, Heidelberg (2010)
8. Bergsten, B., Couprie, M., Valduriez, P.: Overview of parallel architectures for databases. *Comput. J.* 36(8), 734–740 (1993)
9. Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. *Computers and Geo-sciences* 10(2-3), 191–203 (1984)
10. Ciciani, B., Dias, D.M., Yu, P.S.: Analysis of replication in distributed database systems. *IEEE Trans. on Knowl. and Data Eng.*, 247–261 (1990)
11. Cuzzocrea, A.: Theoretical and practical aspects of warehousing, querying and mining sensor and streaming data. *Journal of Computer and System Science* 79(3), 309–311 (2013)
12. DeWitt, D., Madden, S., Stonebraker, M.: How to build a high-performance data warehouse, http://db.lcs.mit.edu/madden/high_perf.pdf
13. Hsiao, H.I., Dewitt, D.J.: Chained declustering: A new availability strategy for multiprocessor database machines. In: ICDE 1990, pp. 456–465 (1990)
14. Coffman Jr., E.G., Leung, Joseph, Y.-T., Ting, D.W.: Bin packing: Maximizing the number of pieces packed 9, 263–271 (1978)
15. Lima, A.A.B., Mattoso, M., Valduriez, P.: Adaptive Virtual Partitioning for OLAP Query Processing in a Database Cluster. In: Lifschitz, S. (ed.) SBBD 2004, Brasilia, Brésil, pp. 92–105 (2004)
16. Lima, A.B., Furtado, C., Valduriez, P., Mattoso, M.: Parallel olap query processing in database clusters with data replication. *distributed and parallel databases. Distributed and Parallel Database Journal* 25(1-2), 97–123 (2009)
17. Loukopoulos, T., Ahmad, I.: Static and adaptive distributed data replication using genetic algorithms. *Journal of Parallel and Distributed Computing* 64(11), 1270–1285 (2004)
18. Menon, S.: Allocating fragments in distributed databases. *IEEE Transactions on Parallel and Distributed Systems* 16(7), 577–585 (2005)
19. Nehme, R.V., Bruno, N.: Automated partitioning design in parallel database systems. In: ACM SIGMOD 2011, pp. 1137–1148 (2011)
20. Pavlo, A., Curino, C., Zdonik, S.: Skew-aware automatic database partitioning in shared-nothing, parallel oltp systems. In: ACM SIGMOD 2012, pp. 61–72. ACM, New York (2012)
21. Rao, J., Zhang, C., Lohman, G., Megiddo, N.: Automating physical database design in a parallel database. In: ACM SIGMOD 2002, pp. 558–569 (June 2002)
22. Stöhr, T., Märten, H., Rahm, E.: Multi-dimensional database allocation for parallel data warehouses. In: VLDB 2000, pp. 273–284 (2000)

Big Data New Frontiers: Mining, Search and Management of Massive Repositories of Solar Image Data and Solar Events

Juan M. Banda, Michael A. Schuh, Rafal A. Angryk, Karthik Ganesan Pillai,
and Patrick McInerney

Montana State University, Bozeman, MT 59717 USA
{juan.banda,michael.schuh,angryk,k.ganesanpillai,
patrik.mcinerney}@cs.montana.edu

Abstract. This work presents one of the many emerging research domains where big data analysis has become an immediate need to process the massive amounts of data being generated each day: solar physics. While building a content-based image retrieval system for NASA's Solar Dynamics Observatory mission, we have discovered research problems that can be addressed by the use of big data processing techniques and in some cases require the development of novel techniques. With over one terabyte of solar data being generated each day, and ever more missions on the horizon that expect to generate petabytes of data each year, solar physics presents many exciting opportunities. This paper presents the current status of our work with solar image data and events, our shift towards using big data methodologies, and future directions for big data processing in solar physics.

1 Introduction

With the launch of NASA's Solar Dynamics Observatory (SDO) on February 11, 2010, researchers in solar physics entered the era of Big Data. SDO is the first mission of NASA's Living With a Star (LWS) program, a long term project dedicated to studying aspects of the Sun that significantly affect human life, with the goal of eventually developing a scientific understanding sufficient for prediction. Space weather (originating from the Sun) is currently considered to be one of the most serious threats to our communication systems, power grids, and space and air travel [1]. Solar storms can interfere with radio communications and satellites (GPS, etc.), and induce geomagnetic currents in our power and communication grids, oil and gas pipelines, undersea communication lines, telephone networks, and railways. A 2008 U.S. government report prepared for the Federal Emergency Management Agency put the yearly financial impact of a massive solar storm event at more than US \$1 trillion (<http://bit.ly/14GjFUJ>).

In the following subsections we will show how the Big Data four V-dimensions of: Volume, Velocity, Variety, and Veracity directly apply to solar data. We highlight several key points on how these dimensions need to be addressed by adapting and expanding our work using and developing big data methodologies.

1.1 Volume and Velocity

The instruments onboard the Earth-orbiting SDO spacecraft currently generate about 70,000 high resolution images (4096x4096 pixels each) per day (Fig. 2a) (VELOCITY), sending back to Earth about 0.55PB of raster data every year (VOLUME). NSF is already in process of building a new ground-based instrument in Hawaii, called the Advanced Technology Solar Telescope (ATST) which is expected to capture about 1 million images per day (3-5PB of data per year).

Currently, the volumes of near-continuous SDO raster data processed all over the world are generating significant amounts of image and object data, and posing significant data mirroring issues related to the distributed character of these massive data repositories. Moreover, many automated computer vision software modules work continuously on this massive data stream to facilitate space weather monitoring. With ATST the amount of data to be processed will be too extreme to be processed in real-time and considerable sampling will need to take place if the current algorithms are not scaled to the task.

1.2 Variety and Veracity

There is a multitude of diverse data about the Sun coming from different instruments and software modules. Ongoing efforts exist to integrate the data under the Virtual Solar Observatory umbrella (<http://bit.ly/18cpyk6>). This situation leads to significant data integration challenges, which are of crucial importance for long-term, solar cycle-oriented (each approx. 11 years) research investigations. The VARIETY of solar data can best be described by two examples: 1) Some of the oldest solar data repositories come from space missions in the 1990s, such as Yohkoh Data Archive Center (<http://bit.ly/1279tsv>), which contains data from a telescope launched by Japan in 1991, and SOHO (<http://1.usa.gov/17v2FaD>), a joint project between the European Space Agency (ESA) and NASA originated in 1995. 2) There is a wide variety of data compacting and meta-data reporting services such as Heliviewer (<http://bit.ly/13imn3i>), and the Heliophysics Events Knowledgebase (<http://bit.ly/14GkWeA>), which provide spatiotemporal data about solar events in vector formats.

Almost all of these resources come from government-funded instruments and/or have data repositories maintained by large companies (e.g. Lockheed Martin) or governmental institutions (e.g. SAO, ESA, NASA). This guarantees high data quality, with certain data standards prototyped over 20 years ago, and assures data VERACITY.

2 Current State of Solar Physics Data Mining

In this section we will cover some of the most important areas of research that the Data Mining Lab at Montana State University (MSU) has been working on over the last several years while closely collaborating with the MSU Solar Physics department, and the Harvard-Smithsonian Center for Astrophysics. Our collaboration started with the objective of building a content-based image retrieval

system (CBIR) for the SDO mission and has developed into new and interesting areas of interdisciplinary research between the two fields. We will outline our three main contributions to the field and mention some of the initial challenges.

2.1 SDO Data Pipeline Details

In Figure 1 we present a high-level overview of the SDO pipeline and the main components that relate to our research purposes, for a more detailed and in-depth discussion of SDO and its data flow, please see [13]. SDO is currently on a geosynchronous orbit with a continuous dual-band data downlink to the ground station in New Mexico. The station’s Data Distribution System is able to hold a rolling 30-day storage window before data goes to Stanford University and the JSOC/NetDRMS for distribution of HMI and AIA image data for science teams to process the data via Lockheed Martin Solar and Astrophysics Laboratory (LMSAL) and Smithsonian Astrophysical Observatory (SAO). While most of the Feature Finding Team Modules process the image data from LMSAL, our Feature Extraction Module for the SDO CBIR system codes runs at SAO. Only a handful of modules run at near-real time latency to provide space weather data for NOAA, while our module runs at a 6 minute cadency. Other science modules run at different cadencies and report to the Heliophysics Events Knowledgebase (HEK) at different intervals. Our SDO CBIR system gathers data from HEK and SAO (image parameter files, headers and thumbnails) on a daily basis. This data gets processed by several data preparation and nearest neighbor table generation/update scripts for it to be visible to users on our web-based front-end. While there are plenty of places where the whole system could be improved for better big data analysis, we are currently focusing on optimizing our n-table update and data preparation scripts using Hadoop-based algorithms. Other potential research areas will be discussed on the following pages.

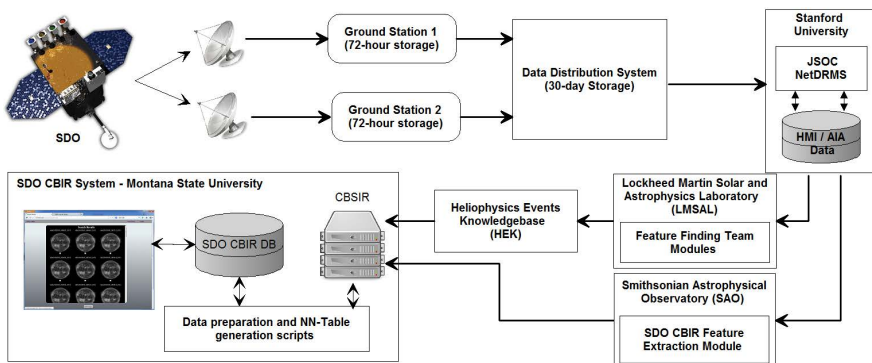


Fig. 1. SDO Pipeline outline

2.2 SDO Content-Based Image Retrieval System

In our task to create a real-world CBIR system for solar data we have faced many interesting challenges from the unique aspects of solar data that relate directly to new and important research questions in computer science and data mining. We have addressed everything from image parameter selection, evaluation, clustering [2,3,4], dissimilarity measures evaluation for retrieval [5], dimensionality reduction analysis for retrieval [6], and evaluation of high-dimensional indexing techniques [7]. We also found very interesting relationships between our solar images and medical x-ray images [8], allowing us to further our research horizons and look into medical image retrieval and CBIR systems [9]. The first version of our system is available at <http://bit.ly/17v3fVG> and currently features over six months of solar data. The system is currently being enhanced with region-based querying facilities and other big data-related enhancements which will be discussed in section 3.2.

2.3 Gathering of Labeled Solar Events from Multiple Sources

The Heliophysics Event Knowledgebase (HEK) is an all-encompassing, cross-mission meta-data repository of solar event reports and information. This meta-data can be acquired at the official web interface <http://bit.ly/ZWdRKH>, but after finding several limitations for large-scale event retrieval, we decided to develop our own software application named QHEK (for Query HEK). Figure 2b shows an example of six types of solar events reported publicly to the HEK from fellow FFT modules. We color-code and overlay the events on the appropriate images (time and wavelength) and show the bounding boxes, and when available the detailed event boundary outlines. A preliminary version of this large-scale dataset is publicly available at <http://bit.ly/15TFTps> and contains over 24,000 event labels from six months of data [10].

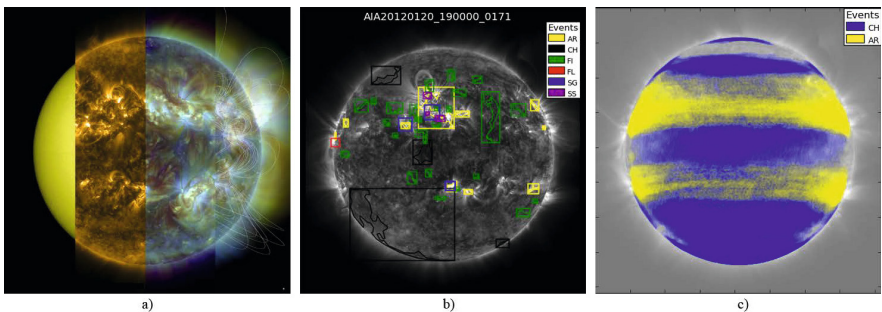


Fig. 2. Examples of SDO solar image data and meta-data. a) courtesy of NASA/SDO

2.4 Visualization of Large Scale Solar Data

To combat serious cases of information overload from the data, meta-data, and results, we have also had to develop extensive visualization tools tailored to our specific data domain and research applications. While not all of our work is directly visual, such as high-dimensional indexing techniques and spatio-temporal frequent pattern mining [7,11,12], almost all of it is related to some sort of visualizable end result. For example, with the help of visualization we can quickly analyze hundreds of solar events at once and validate a module's reporting effectiveness against known solar science, such as the confirmed distinct bands of active regions and coronal holes shown in Fig.2c. We can also use visualization to more easily assess the strengths and weaknesses of our own classification algorithms and labeling methods, whereby the human eye can keenly pick up on similar miss classified regions or poorly generated data labels.

3 Transitions into Big Data Analysis for Solar Physics

The following subsections will give some insights into our work of transitioning from traditional large-scale image retrieval and data mining approaches to big data methodologies and technologies. We also point out several of the research challenges, practical applications of current big data technologies, and the development of new big data analysis algorithms.

3.1 State of the Art in Large Scale Image Retrieval

Large scale image retrieval has been an active topic of research since the late 2000's with the likes of Google Image Search and systems like QBIC. These systems have since become closed, and in their infancy handled mostly meta-data based image search and basic color histogram matching, making them not well suited for current large scale image retrieval needs – a in depth review on CBIR could be found here [2]. With interesting works dealing with more than 10,000 image categories [16] and high-dimensional signature compression for large scale retrieval [17], it is not until 2012 where in the Neural Information Processing Systems (NIPS) conference we find the first Workshop on large scale Visual Recognition and Retrieval. Here researchers presented several algorithms that work on large scale image datasets, but almost none of them mention the use of big data technologies such as Hadoop, HBase, or HSearch. The first real mention of using big data technologies for image retrieval is in [15], where a highly speculative system using Hadoop and Lucene is proposed. We have yet to find literature with a functional system using said technologies. While most of the image retrieval algorithms have been parallelized and tested in distributed environments, either GPU or using OpenMP, they have yet to be ported to Hadoop-based environments. For the future version of our SDO CBIR system we are working on developing a Hadoop-based algorithm for nearest-neighbor index generation.

3.2 Towards Big Data Revision of the SDO CBIR System

Our first step is to verify the feasibility of migrating our traditional SDO CBIR system to a more flexible search-engine based technology using Lucene. With this initial step underway, if successful, we may then migrate to Apache's HBase hadoop-based technology for scalability with the larger amounts of data we will accumulate. We are also looking into incorporating HSearch on our HBase data repository to serve data queries for the front-end of our system. We are deploying scripts to update our CBIR system similarity indexes using MapReduce to calculate and re-calculate our similarity tables on a weekly, and eventually daily, basis in order to provide the most up-to-date results for solar scientists when important events happen (e.g. big solar flares).

The biggest research potential of our current work is the combination of image retrieval, information retrieval, and big data methodologies to create a big data content-based image retrieval system, something that will greatly benefit other areas that are starting to deal with high volumes of image data, but are currently stuck with traditional approaches. We are excited about future collaborations with image processing and retrieval researchers in expanding existing algorithms and methodologies into big data environments.

3.3 Event Labeling Module Validation

With the massive amounts of label data coming from multiple science modules, there is a need for big data technologies capable of aggregating and validating data. The current best existing computer vision tools for labeling solar images are single-object detectors, each heavily reliant on the known visual characteristics of their specific phenomenon for accurate labeling [13]. Object recognition and classification based on more general visual parameters is still limited, although some success has been seen in filament detection [14].

The development of these specific modules is expensive in terms of time, effort and domain knowledge, therefore a general purpose computer vision tool is much better suited for extension to include new phenomena, or to classify subtypes of known phenomena by visual character. For these reasons we seek to develop a tool capable of using the image texture parameters to label and classify events in solar imagery. In this environment we endeavor to construct and test a multi-label event classification scheme for solar images. A major advantage of multi-label classification is that it is known that the occurrences of solar phenomena are not independent. For example, active regions are areas of high solar activity, while coronal holes are areas of low solar activity, so they should never occur in the same location (again, see Fig.2c).

3.4 Spatio-temporal Solar Event Reporting and Mining

Spatio-temporal analysis of solar physics data is a major emerging area and the volume of data this will generate must be addressed using big data analysis methodologies. In our initial stages we are working on establishing an all-encompassing infrastructure in order to store all reported events. The current

reporting involves a single spatial label per temporal event, an event that could range from minutes to days. We are proposing to create tracking datasets with each spatial label converted to a temporal step of our solar data, exploding our dataset from thousands of records to millions. In the SDO data context, we are looking at over 70,000 images with multiple spatio-temporal labels per day.

In our second step, we are investigating the migration of data into HBase, with highly-scalable search capabilities using HSearch. This will be taking advantages of Hadoop/MapReduce environments to process and analyze the data with their clustering and mining algorithms, as well as having a front-end to serve the data for other research institutions. New algorithms will need to be developed to fit the context of spatio-temporal data analysis for big data sources.

4 Looking into the Future

As the volume of solar data keeps growing each day, the transition from using traditional data mining, machine learning, and information retrieval techniques into more scalable big data methodologies and tools is imminent. While we have outlined some of the steps we are currently taking to address these issues, we are also looking for new collaborations with big data experts to further benefit the field of solar physics. As we have shown, there are plenty of new areas of research that can be benefitted from the massive solar datasets and the new tools and algorithms expected to be developed for this domain can be greatly beneficial for other big data research areas.

References

1. Hapgood, M.A.: Towards a scientific understanding of the risk from extreme space weather. *Advances in Space Research* 47(12), 2059–2072 (2011)
2. Banda, J.M., Angryk, R.: Selection of Image Parameters as the First Step Towards creating a CBIR System for the Solar Dynamics Observatory. In: *Proc. of Int. Conf. on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 528–534 (2010)
3. Banda, J.M., Angryk, R.: An Experimental Evaluation of Popular Image Parameters for Monochromatic Solar Image Categorization. In: *Proc. of the 23rd Florida Artificial Intelligence Research Society Conf.*, pp. 380–385 (2010)
4. Banda, J.M., Angryk, R.: On the effectiveness of fuzzy clustering as a data discretization technique for Large-scale classification of solar images. In: *Proc. IEEE International Conference on Fuzzy Systems*, pp. 2019–2024 (2009)
5. Banda, J.M., Angryk, R.: Usage of dissimilarity measures and multidimensional scaling for large scale solar data analysis. In: *Proc 2010 Conf. on Intelligent Data Understanding (CIDU)*, pp. 189–203 (2010)
6. Banda, J.M., Angryk, R., Martens, P.C.H.: On Dimensionality Reduction for Indexing and Retrieval of Large-Scale Solar Image Data. *Solar Phys.* 283, 113–141 (2012)
7. Schuh, M.A., Wylie, T., Banda, J.M., Angryk, R.A.: A comprehensive study of iDistance partitioning strategies for k NN queries and high-dimensional data indexing. In: *Gottlob, G., Grasso, G., Olteanu, D., Schallhart, C. (eds.) BNCOD 2013. LNCS, vol. 7968*, pp. 238–252. Springer, Heidelberg (2013)

8. Banda, J.M., Angryk, R., Martens, P.: On the surprisingly accurate transfer of image parameters between medical and solar images. In: Proceedings of the International Conference on Image Processing (ICIP), pp. 3730–3733 (2011)
9. Müller, H., Michoux, N., Bandon, D., Geissbuhler, A.: A review of content-based image retrieval systems in medical applications: clinical benefits and future directions. *International journal of medical informatics* 73, 1–23 (2004)
10. Schuh, M.A., Angryk, R.A., Pillai, K.-G., Banda, J.M., Martens, P.C.H.: A large-scale solar image dataset with labeled event regions. To appear in. In: Proc. of the International Conference on Image Processing, ICIP (2013)
11. Pillai, K.-G., Angryk, R.A., Banda, J.M., Schuh, M.A., Wylie, T.: Spatio-temporal co-occurrence pattern mining in data sets with evolving regions. In: *ICDM Workshops 2012*, pp. 805–812 (2012)
12. Pillai, K.G., Sturlaugson, L., Banda, J.M., Angryk, R.A.: Extending high-dimensional indexing techniques pyramid and $i\text{MinMax}(\theta)$: Lessons learned. In: Gottlob, G., Grasso, G., Olteanu, D., Schallhart, C. (eds.) *BNCOD 2013*. LNCS, vol. 7968, pp. 253–267. Springer, Heidelberg (2013)
13. Martens, P.C.H., Attrill, G.D.R., Davey, A.R., Engell, A., Farid, S., et al.: Computer vision for the solar dynamics observatory (SDO). *Solar Physics* (2011)
14. Schuh, M.A., Banda, J.M., Bernasconi, P.N., Angryk, R.A., Martens, P.C.H.: A comparative evaluation of automated solar filament detection. *Solar Physics* (under review, 2013)
15. Gu, C., Gao, Y.: A Content-Based Image Retrieval System Based on Hadoop and Lucene. In: *Cloud and Green Computing (CGC)*, November 1-3, pp. 684–687 (2012)
16. Deng, J., Berg, A.C., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part V*. LNCS, vol. 6315, pp. 71–84. Springer, Heidelberg (2010)
17. Sánchez, J., Perronnin, F.: High-dimensional signature compression for large-scale image classification. In: *Proc. of CVPR* (2011)

Extraction, Sentiment Analysis and Visualization of Massive Public Messages

Jacopo Farina, Mirjana Mazuran, and Elisa Quintarelli

Politecnico di Milano,
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
jacopo1.farina@mail.polimi.it, {mazuran,quintare}@elet.polimi.it

Abstract. This paper describes the design and implementation of tools to extract, analyze and explore an arbitrarily great amount of public messages from diverse sources. The aim of our work is to flexibly support sentiment analysis by quickly adapting to different use cases, languages, and message sources. First, a highly parallel scraper has been implemented, allowing the user to customize the behavior with scripting technologies and thus being able to manage dynamically loaded content. Then, a novel framework is developed to support agile programming, building and validating a classifier for sentiment analysis. Finally, a web application allows the real-time selection and projection of the analysis results in different dimensions in an OLAP fashion.

Keywords: sentiment analysis, big data, OLAP analysis.

1 Introduction

The increasing usage of networks and the improvement of communication technologies have made it possible to access a previously unimaginable amount of messages written by the general public. Companies can exploit these messages for getting an insight into the opinion the public has about their products and brands. This opinion, called sentiment, ranges from negative to positive in various forms and grades, therefore, given a scenario, it is necessary to identify a set of possible classes of sentiments that should be used by a classifier in order to automatically associate sentiments to each message. In the following we will adopt three classes: *positive*, *negative* and *neutral*.

The discipline that studies the design and implementation of tools able to automatically detect the sentiment of a text is called *sentiment analysis* [12]. It is interesting to detect how the sentiment changes over time, in different geographical areas and within different keywords to focus on specific aspects of the products. Moreover, it is useful to get an insight about the differences in sentiment from various sources, for instance the comments of YouTube videos or FaceBook pages, hence allowing a company to find the ones having the highest impact in terms of number of messages and expressed sentiment.

Differently from [14] where a limited stream of comments is analyzed in real time, our goal is to efficiently analyze a great amount of comments written in a

wide time span, in some cases more than 10 years. Since the amount of available messages can be huge and is rapidly growing year after year, it is necessary to resort to the use of techniques exploiting networks of computers able to process in parallel several distinct portions of the overall input data.

Although recent technologies like *MapReduce* [1] facilitate the development of distributed applications managing the most common issues emerging from parallel environments, the implementation of these applications is still more difficult than non-distributed ones, because of the great effort and time consumption required for detecting problems and errors, finding the corresponding corrections and applying them to the different nodes.

A sentiment analysis classifier, conversely, requires some sort of agile development to quickly try variations and improvements of the classification algorithm and validate the results. Thus, in this paper we introduce a framework enabling developers to run the same application both in a local and in a distributed environment. In the first case it is possible to quickly test and validate the application on a small input data sample, while in the second case the application is able to manage great amounts of user messages by scaling linearly.

Furthermore, the application allows real-time selection and graphical representation of millions of messages in the various dimensions (time, place, keywords, sources and sentiment), features that require an accurate selection and adaptation of specific technologies.

The structure of the paper is as follows: in the next section we introduce the adopted technologies and their state of the art; in Section 3 we explain the architecture of our system. In Section 4 we show how our system can be adopted on a single node and finally, in Section 5 we draw the conclusions of our work.

2 State of the Art and Adopted Technologies

Web Scraping. While most social networks offer APIs to allow the extraction of messages, web forums usually consist in dynamic web pages that require the use of a *scraper* to extract data. A scraper is a tool that downloads pages and follows HTML links inside them. This task can be very time-consuming since a site can consist of millions of pages, so scrapers use parallelization to fetch many pages simultaneously and filters to avoid unwanted pages. In some cases, web pages contain scripts, run by the browser, which trigger the loading of the actual content; in this cases, a scraper will not extract it since it does not run page scripts. Running all of the page scripts requires to implement or adapt a very complex software and makes scraping several degrees of magnitude slower.

Currently, the most sophisticated tool is *Apache Nutch*, a distributed scraper, which can scale linearly, increasing the number of clusters and building an index of one or more websites. However, it is aimed to index pages for further searches and not to extract specific data. Moreover, there is a limit on the speed of page extraction from single websites making pointless such an heavy solution, while dynamic content, loaded with AJAX, is not considered.

MapReduce-Based Distributed Computing. As for distributed processing, MapReduce is a paradigm for the development of distributed applications through the definition of two functions, *map* and *reduce*. They allow the programmer to easily distribute the application on multiple nodes, not worrying about which node will process which chunk of input data and reassigning work and data in case of node failure.

In particular, a map function takes in input a key-value pair $\langle k_1, v_1 \rangle$, and outputs a list of key-value pairs $\langle k_2, v_2 \rangle$, which may be empty. A reduce function receives in input a key k_2 and the list of associated values $\{v_2, v_3, v_4, \dots\}$ produced by the map function, generating another set of pairs $\langle k_3, v_5 \rangle$, which is the result of the process and could be further used by other MapReduce operations. Sometimes one of the two function is the identity function, i.e. it can be omitted. For example, a map function could be used as a filter to produce only pairs corresponding to some criteria, thus the reduce function will be omitted and the result of the job will be the filtered input data.

Apache Hadoop is the state-of-the-art implementation of the MapReduce paradigm which implements a distributed filesystem, *HDFS*, that automatically divides files in blocks and stores different copies of each block in different nodes. Using Hadoop, a MapReduce *job*, consisting of a map or reduce function, is divided in *tasks*, which are assigned to the nodes by a central coordinator called *JobTracker*. Hadoop aims to reduce the network traffic assigning tasks to the nodes owning the chunks to process; in case of failures or excessive delays, however, the same task is assigned to distinct nodes and chunks are replicated to avoid data loss.

While greatly leveraging the application development by managing common distributed applications problems, the deploying and execution of them on Hadoop clusters is very slow when the data is relatively small, due to heavy initialization procedures and data redundant replication.

Sentiment Analysis. *Sentiment Analysis* is the possibility to automatically classify the mood expressed by a document and, in some cases, the subject of emotionally expressive utterances [12]. Various models and classifiers can be used, in general SVMs and Bayesian classifiers give the best results [13], the training of the latter type can be described as a MapReduce operation, hence it was chosen for this use case.

A statement can express various sentiments, directed toward distinct subjects or distinct features of the same subject (e.g. "*Milan is beautiful but the weather is terrible*"). It is very difficult to automatically identify these elements inside statements, so most of the approaches to sentiment analysis try to detect the sentiment expressed in the whole text [16] assuming that texts in general have one main subject [11], and under this assumption the application detect the mood of a text as a whole, ignoring distinct cited subjects.

Currently, there are not widely used tools nor datasets for sentiment analysis, due to the variety of possible definitions of the task. On the other hand, scientific literature regarding the models suitable for sentiment analysis is rich [5] [12].

However, languages different than English are less often subject of work and coped using automatic translation [4], which gives poor results when applied to internet language [6]. Various approaches exploit databases of words expressing sentiments, Sentiwordnet, which provides a list of token polarized in the three classes (positive, negative and neutral) is the most similar to the proposal of this paper. However, in our application scenario this is not feasible, since handmade datasets are limited to one or a few languages and suffer of a low accuracy.

3 System Architecture

Our system consists of the chain shown in Figure 1, where each oval represents data and each box describes operations performed on the data.

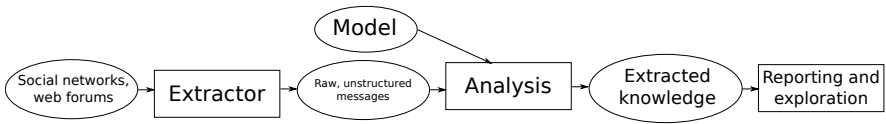


Fig. 1. System architecture

Intuitively, our system takes as input a defined set of data sources (such as data coming from social networks, forums and so on) and produces, through the *extractor* component, raw unstructured text messages. These messages are the input of the *analysis* component that generates knowledge by enriching each message with: (i) sentiment and (ii) geographical provenience. This analysis is supported by a probabilistic model of sentiment, which uses a set of previously classified messages, i.e. messages assigned by hand to sentiment classes with a certain probability, in order to automatically classify new ones. The described knowledge is then stored along with the original text as indexed structured data and is used as input for the *reporting and exploration* component of the chain. This component allows the final user to run queries over the data: it supports common OLAP operations such as drill down and slice and dice.

The proposed system can be seen as a middle ground between an ETL and an OLAP tool. The first two components in our framework perform ETL operations, that is, extract data from web sources, transform it in appropriate knowledge including sentiments, and finally load it into a component that implements OLAP operations over the data. The three components are implemented for users with little or no knowledge of sentiment analysis, distributed processing technologies and their methods. The OLAP interface, in particular, is entirely graphical, allowing non expert users to explore data autonomously.

To validate our approach, we consider web forums that contain millions of messages on a specific topic, written over time by a huge amount of users from different places. Companies distributing a product or service related to that topic are interested in getting an overall view of the opinions among users, extracting relevant trends and relations between product names, places, keywords and,

above all, sentiments. Not knowing beforehand which relations and trends express useful knowledge, our approach is to classify and index data along these dimensions and let the user freely explore and select through an OLAP-like analysis. From a web forum containing, among others, the message:

Seen the last model yesterday at Chicago, not so exciting :(

we need to traverse pages, extract the message text from the HTML page within the publication date, detect the reference to *Chicago*, the negative sentiment, the title of the topic. Using these data we can for example allow the user to get a visual representation of how many negative posts were written in Illinois over the last 3 years, grouped by month, with an immediate query response even in case of millions of data points.

3.1 Extraction

The extractor component consists in a scraper that has been implemented intensively exploiting multithreading and customization through Mozilla Rhino¹ scripting. The component allows users to quickly redefine the behavior of the tool, for example, by setting to avoid or redirect the URLs that have to be examined, or to select and extract only small sections of the pages or yet, to download dynamic content. This leads to a rapid but strong customization of the tool behavior. This feature is usually not allowed by existing tools (such as *httrack*² and *Apache Nutch*³) that generally allow only to specify URLs that have to be avoided or followed using regular expressions. Moreover, dynamically loaded content, like AJAX⁴, cannot be easily extracted.

Our scraper has a default behavior which consists in a graph search leading to download all the pages and following all the links until all URLs found in the traversal are examined. When the scraper downloads a page, the XHTML code is parsed and the compiled script is called, leaving to the user the task to define which elements are to be extracted, if any, and which URLs are to be followed whether the user needs to avoid the default behavior of following all the hyperlinks. The user script can enqueue arbitrary addresses to simulate AJAX calls, hence allowing to manage dynamically loaded content, and JSON annotations can be added to each pending URL allowing the user script to maintain a context during page traversal.

A web forum can thus be parsed efficiently defining a script usually shorter than 20 lines and extracting messages in this form:

```
{ "post": "Seen the last model yesterday at Chicago, not so
  exciting :(", "date": "2013-02-16 10:34:00" }
```

¹ <https://developer.mozilla.org/en-US/docs/Rhino>

² <http://www.httrack.com>

³ <http://nutch.apache.org>

⁴ [https://en.wikipedia.org/w/index.php?title=Ajax_\(programming\)&oldid=553459243](https://en.wikipedia.org/w/index.php?title=Ajax_(programming)&oldid=553459243)

The JSON string contains the raw text and the date of the post, and will be used by the next steps. The tool uses a Bloom filter [2] to keep the examined URLs list in volatile memory thus scaling up to millions of pages examined on commodity hardware without accessing the disk, at the cost of a small (e.g. $< 10^{-8}$) and predictable ratio of ignored valid URLs. Thanks to the use of this filter, the application can be deployed, monitored and modified on a laptop and updated in real time fastening the procedure, which can last various days for internet forums.

3.2 Analysis

The analysis component takes as input the raw data collected by the extractor and enriches it with sentiment. First, text is put in lowercase and some patterns (money amounts, percentages, numbers, emoticons, etc.) are replaced with placeholders, obtaining for instance:

```
seen the last model yesterday at chicago , not so exciting
SMILENEG
```

where "*SMILENEG*" is a placeholder for the negative smile. Text can be subjected to stemming to increase accuracy, thus a heuristic stemmer [18] was implemented and trained using a large amount of messages from a web forum. For instance, the previous statement can possibly become:

```
se the last model yesterda at chicago , not so excit SMILENEG
```

where the common suffix has been removed. The fact that the word "*yesterday*" lost the ending *y* is a mistake from a grammar point of view but correct with respect to the goal of removing conjugations, since no other word becomes "*yesterda*" after stemming.

Since a naïve model gives poor results with aggregate terms, tokens within a distance k are merged to form a richer feature vector and take into account whole expressions like "*I don't like*" and not only single tokens. The statement is thus divided in tokens using spaces and punctuation as token separator, then aggregate tokens are added, obtaining:

```
[se , yesterda , ... , SMILENEG , se_1_yesterda , ... ,
excit_1_SMILEPOS , se_2_at , ... ,so_2_SMILENEG]
```

where "*so_2_SMILENEG*" represents the aggregate term made by "*so*" followed by *SMILENEG* after another token. Through this operation structures like "*not_2_exciting*" can be maintained and considered in further steps. Now, the original message has become a set of tokens and thus can be treated as a boolean vector indicating the presence or absence of tokens. Using a set of statements labeled by hand with a sentiment, like:

```
{"post":":":(" , "sentiment":" negative" }
{"post":":":Seen the last model yesterday at Chicago , not so
exciting :(" , "sentiment":": negative" }
{"post":":":This book is short , but very exciting" , "sentiment":":
positive" }
```

a list of tuples $\langle token, sentiment \rangle$ is generated, and the occurrences of each token, in the different sentiment classes, is counted:

```
{ "token": "SMILENEG", "negative": 2 }
{ "token": "book", "positive": 1 }
{ "token": "excit", "negative": 1, "positive": 1 }
{ "token": "not_2_excit", "negative": 1 }
{ "token": "ver_1_excit", "positive": 1 }
...
```

Note that, although the token *excit(ing)* occurs in both sentiment classes, only "not_2_excit" is associated with a negative sentiment.

Next, Laplacian smoothing is introduced to contrast overfitting [19], that is, all the counters for all the tokens and classes are increased by 1. After smoothing, the polarity of each token for each class is calculated as the ratio of occurrences for that class on total occurrences. For example:

```
{ "token": "excit", "negative": 0.40, "positive": 0.40, "neutral": 0.20 }
{ "token": "not_2_excit", "negative": 0.50, "positive": 0.25, "neutral": 0.25 }
{ "token": "ver_1_excit", "negative": 0.25, "positive": 0.50, "neutral": 0.25 }
...
```

Therefore, using a set of messages assigned to sentiment classes by hand, for each class i , a vector M_i assigning a polarity to each token has been generated. The procedure to transform a previously unseen text in a vector is the same, but instead of considering token polarities, only the presence is taken into account, thus generating a vector S containing only 0 or 1. A token occurring multiple times in the same text is counted only once. The dot product $S \cdot M_i$ gives a likelihood score composed by a text vector and a sentiment class vector. The sentiment class giving the highest score for a text is chosen as the sentiment class of it. This likelihood is not normalized, that is, the sum of likelihoods for all the classes is not 1, because a common normalizing positive factor has been ignored being irrelevant to the comparison of classes.

Once the polarities of tokens have been calculated from the manually classified messages, it is possible to apply them to classify new text by multiplying polarities of found tokens and assigning to the sentiment with the greatest result, obtaining a naïve Bayes classifier. The classifier has been tested on various datasets of messages from Twitter, Facebook, three web forums and YouTube, in Italian and English, or various mixed languages in the case of YouTube comments, trained and validated using the framework through k -folding with different values of k and with or without the stemming, obtaining a precision, measured as the ratio of results matching with an human classifier between 0.42 and 0.62. It has to be noted that human evaluators has a disagreement rate for this task which can reach 0.4 on short texts [15].

3.3 Reporting and Exploration

Datasets are indexed using a relational database, which can scale linearly on a number of nodes under some conditions [3], namely PostgreSQL. A datacube of analyzed and indexed messages has been built as an immutable data structure; slicing operations produce brand-new datacubes, allowing many users to slice them without interfering with each other.

Our system provides a web application that allows users to filter a datacube specifying time ranges, keywords, sentiments, place names, rectangles of geographical coordinates and any combination of these. These filters generate a new datacube which can be further analyzed and filtered. A datacube can be represented along all the dimensions using specific representations (geographical heatmaps, sentiment histograms and pie charts, list of messages colored in accordance with sentiment, YouTube videos, FaceBook messages and forum threads with the most positive or negative sentiment of comments), intensively using HTML5 and AJAX to display data in an engaging and interactive application that is shown in Figure 2. This application is able to group data in real time, for instance a user can see sentiment trend histograms grouped by months, and years (e.g. all the posts made in the month of january over the years) and at different time resolutions ranging from hours to years, just clicking links without reloading the page.

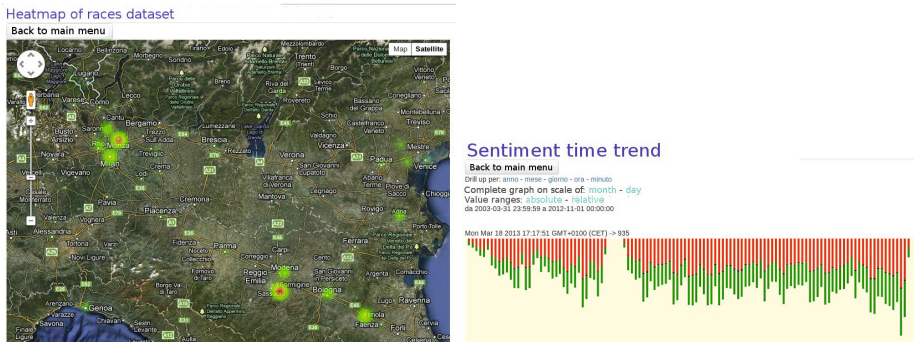


Fig. 2. A heatmap, showing the density of messages across different places, and an histogram showing the trends of sentiments across time. Both views are interactive, allowing the user to zoom, pan and change intervals, calculating results in real time.

For each slicing operation, the application generates and displays both the SQL query and the workflow code to run the same operation using Hadoop⁵ and the described framework, and allows the user to execute it directly. Another feature of the web application is the possibility to visit immediately the sources of messages, like viewing the videos of YouTube from where the comments were extracted.

We remark that, to avoid conflicts between multiple simultaneous users, datacubes are immutable: the result of a slicing operation is a new datacube, which can in turn be subject of other slicing operations, leaving untouched the original

⁵ <http://hadoop.apache.org>

dataset and allowing the rapid visualization of different types of charts on the same selection.

4 Framework for Agile Development

The same paradigm of MapReduce, involving an abstract description of the operations on data and leaving to an external application the task of distributing jobs and data among the nodes, has been extended; it abstracts the existence of a cluster at all, allowing a programmer to define a JavaScript workflow which will call a few generic operations or define its own through scripting.

The workflow includes various *phases*, consisting of a map or reduce task (or both), specifying the input and output files and optionally some execution parameters in the form of a JSON object. Each phase defines map or reduce functions as operations over JSON keys and values. This allows the management of unstructured data; indeed these objects are seen as textual keys and values by Hadoop, but can encapsulate dictionaries of arbitrary keys and values and nested objects. Moreover, this format is human readable and can be efficiently compressed, easing the inspection of partial results without wasting disk space.

This workflow can hence be run seamlessly on a single node with a local file system or in a distributed environment using Hadoop, enabling agile development. Through an intense use of scripting and JSON format, it is possible to rapidly define and test in local mode a workflow to manipulate unstructured data and deploy it to the cluster nodes without changes.

In case of Hadoop execution, the framework will embed phases in actual Hadoop tasks, converting JSON strings in Hadoop textual key-value pairs; in local mode the framework will read files, extract JSON objects and pass them to the phases, writing emitted output in the local file system and to the console allowing the user to monitor the application in real time.

5 Conclusion

In this work we have presented a system for the extraction, sentiment analysis and visualization of huge amounts of public messages. Our system is composed of: i) a highly customizable scraper that extracts data from different sources; ii) an analyzer that adds sentiment to the extracted data using a probabilistic model and iii) a web application that allows non-expert users to query the obtained knowledge through OLAP operations and visualize the results.

Acknowledgements. This work was partially funded by the Italian project Sensori (Industria 2015Bando Nuove Tecnologie per il Made in Italy) Grant n. 00029MI01/2011.

References

1. Pike, R., Dorward, S., Griesemer, R., Quinlan, S.: Interpreting the Data: Parallel Analysis with Sawzall. Special Issue on Grids and Worldwide Computing Programming Models and Infrastructure 13(4), 227–298

2. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM* (1970)
3. Yang, C., Yen, C., Tan, C., Madden, S.R.: Osprey: Implementing MapReduce-style fault tolerance in a shared-nothing distributed database. In: *ICDE*, pp. 657–668 (2010)
4. Bautin, M., Vijayarenu, L., Skiena, S.: International sentiment analysis for news and blogs. In: *ICWSM* (2008)
5. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* (2008)
6. Clark, E., Araki, K.: Text normalization in social media: progress, problems and applications for a pre-processing system of casual English. *Procedia-Social and Behavioral Sciences* 27, 2–11 (2011)
7. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs Up? Sentiment Classification Using Machine Learning Techniques. In: *Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing*, vol. 10 (2002)
8. Snyder, B., Barzilay, R.: Multiple Aspect Ranking using the Good Grief Algorithm. In: *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference* (2007)
9. Pang, B., Lee, L.: Seeing Stars: Exploiting Class Relationships For Sentiment Categorization With Respect To Rating Scales. In: *Proceedings of ACL*, pp. 115–124 (2005)
10. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: *Proceedings of LREC*, pp. 417–422 (2006)
11. Meena, A., Prabhakar, T.V.: Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In: Amati, G., Carpineto, C., Romano, G. (eds.) *ECiR 2007. LNCS*, vol. 4425, pp. 573–580. Springer, Heidelberg (2007)
12. Nasukawa, T., Yi, J.: Sentiment analysis: Capturing favorability using natural language processing. In: *Proceedings of the 2nd Int. Conference on Knowledge Capture*, pp. 70–77. ACM (2003)
13. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D.: Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 61(12), 2544–2558 (2010)
14. Ahmad, K., Cheng, D., Almas, Y.: Multi-lingual sentiment analysis of financial news streams. In: *Proc. of the 1st Intl. Conf. on Grid in Finance* (2006)
15. Gill, A.J., Gergle, D., French, R.M., Oberlander, J.: Emotion Rating from Short Blog Texts. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008)
16. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347–354. Association for Computational Linguistics (2005)
17. Chang, P.-C., Galley, M., Manning, C.D.: Optimizing Chinese word segmentation for machine translation performance. In: *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 224–232. Association for Computational Linguistics (2008)
18. Porter, M.F.: *Snowball: A language for stemming algorithms* (2001)
19. Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: *Proceedings of the 18th International Conference on Machine Learning*, pp. 609–616 (2001)

Desidoo, a Big-Data Application to Join the Online and Real-World Marketplaces*

Daniele Apiletti and Fabio Forno

Ooros, via Legnano 27, Torino, Italy
{daniele.apiletti,fabio.forno}@ooros.com

Abstract. The paper presents a big-data application in the context of an innovative marketplace service running in the cloud. The marketplace aims at bridging the gap between the online e-commerce world and the offline physical places and shops. Experiences from the system startup, design patterns and challenges to scale the platform are discussed.

1 Introduction

In recent years, a strong trend to run most ICT services in the cloud has emerged [1], while time spent by users online has seen a sharp increase, in particular on mobile devices [2]. Such devices bring the user online when away from the traditional seat at desk, enabling access and consumption of digital contents on the go in the real world.

Most small and medium businesses (e.g., shops) have an established offline presence in the real world but fail to engage with their customers online. Such businesses strive to create, manage, and exploit a digital presence, which typically consists of websites, pages on social networks, and e-commerce platforms: entirely different worlds with respect to the established offline shop, where their main business activity takes place.

While some businesses are based or can live solely online, others definitely need a real-world presence to fully engage their customers. Hence the need for a new big-data application to:

- Store, process, and extract actionable knowledge from the huge amount of information of online and offline business activities and their customers.
- Bridge the gap between the online and offline worlds, where small and medium businesses can benefit from both a storefront presence and a digital shop window.
- Connect customers and merchants not only at the sale event, but reinterpret the new trends in social networking by establishing long-lasting business relationships among the business actors, merchants and customers, that each other can benefit from.

* The views expressed in this article are solely of the authors and do not necessarily represent those of the company or of its board.

- Support business owners in focusing on their job, by automating a series of activities related to the Customer Relationship Management through the platform in the cloud and directly in the shop.

Such big-data application is now live as an innovative marketplace, it has been fully forged by a Turin-based start-up in Italy, and it is currently in an early public stage under the commercial name of Desidoo [3].

The challenges faced to design and develop from scratch this big-data platform to allow scaling the marketplace, experiences, and design patterns of the application are presented in the paper, such as the heterogeneous data model, replication and sharding, aggregate statistic computation, and real-time analytics.

The paper is organized as follows. Section 2 presents the marketplace architecture and a functional overview of the platform. Section 3 describes design choices and challenges faced while developing the application. Section 4 draws conclusions and discusses improvement directions.

2 Application Architecture

This Section provides an overview of the platform architecture in terms of functionality and building blocks.

2.1 Functional Overview

The system actors (customers and merchants) can access the marketplace from three different channels: (i) the website, (ii) the mobile app, and (iii) in the shop by means of a fully customized touch-enabled tablet.

The tablet is the actual device connecting the real-world shop to the online digital marketplace. It is designed to run in the cloud, so that in case of failure, replacement, or relocation, no data are lost and a new login on the merchant account makes it quickly operational again. The back-end acts as a data store, and handles all the data crunching tasks, presenting results to the tablet. The tablet, however, gracefully handles some offline activities to prevent internet connection failures from stopping in-shop service availability.

The tablet is equipped with an NFC reader, that allows a physical interaction with an RFID chip. This feature is exploited by providing customers RFID-enabled keyrings, which become their keys to access the marketplace in the real-world shops.

Customers can link their RFID keyrings to their online user account on the marketplace. This action virtually pulls on the linked keyring the online customer profile, her preferences, special offers, e-commerce products, fidelity points, and her whole online history, enabling a deep personalization of the tablet interactions: the marketplace knows the customer directly in the shop, and can welcome her with a personalized photo, name, messages, particularly targeted offers, reserved specials, and so on. Hence, customers can buy online and redeem offline, while in the shop they can select offers from the online marketplace directly

through the tablet interaction, and get fidelity points for purchases. All actions in both worlds are synchronized in real-time when the tablet is connected, or asynchronously when the tablet is offline.

From a merchant point of view, besides common business features such as creating promotions, managing their redemption, contacting customers, customizing fidelity programs, etc., a new set of business-relevant metrics can be collected and exploited to increase customer loyalty and drive purchases, such as the frequency of shop visits per customer, the number of visits leading to purchases, new and recurrent customer segmentations. Furthermore, set-and-forget triggers can be fully customized to free the merchant from the hassle of handling customers' birthdays, expiring offers, welcome messages, and other repetitive tasks. Finally, new customers can be brought from the online world into the real shop, and co-marketing actions with local nearby shops can be easily launched, thus fostering new business channels and relationships, and empowering local merchants in the neighborhood.

A complete description of the marketplace features for customers and merchants is beyond the scope of the paper, and would be outdated due to the quick evolution of the system itself, however a basic list of concepts are introduced in Section 2.2 to allow presenting challenges and experiences in designing the big-data application.

2.2 Transactions

The core of the platform consists of events and activities on the marketplace. Such data are modeled as transactions. Transactions are stored in the back-end database with a combination of some common attributes (source, destination, type, timestamp, etc.) and additional variable fields depending mainly on the event type, e.g., the number or amount, whose meaning and units of measure depends on the specific transaction.

Additional fields can also be added and indexed on purpose (e.g. list of tags), to speedup the search of events that become particularly relevant due to special temporary marketing operations or new permanent business strategies.

A grouping operation is then performed on the transactions to compute totals over specific dimensions, such as per customer, per shop and over time. Details on this task are reported in Section 3.1.

Some relevant transactions are described in the following.

- **Subscriptions.** When a customer visits a shop and check-ins through her RFID keyring on the tablet, a pub-sub¹ subscription is created to allow updates from the shop to reach its customers, possibly in real-time (e.g., app notification, website message, SMS, email).
- **Customer bonuses.** A set of bonuses are linked to the customer profile, from shop-specific fidelity points assigned by merchants for purchases, to system-wide credits rewarding active customers on the marketplace, from

¹ Publish subscribe, a pattern to efficiently dispatch contents, typically messages, from producers to readers.

badges of marketplace competitions, to special statuses of VIP customers in specific shops.

- **Offers and products.** The most popular contents on the marketplace, besides customer and shop profiles, are the offers and products on sale. The marketplace rewards customers with credits for buying products. They can do so online, and get products shipped at home or go to the shop and collected them after checking-in with their RFID keyring. By spending credits, customers can book offers, online or directly in the shop. The payment is requested only at the redemption in the shop. Publishing, blocking, expiring, booking, paying, collecting, and redeeming offers or products are some of the transactions involved in managing such contents.
- **Triggered events.** The marketplace provides merchants with automatic recipes for their businesses. Some of these are welcome offers for new customers, cross-welcome offers for customers of other merchants (typically located in the neighborhood), fidelity rewards when reaching a given amount of fidelity points, etc. All these actions are triggered by one or more transactions among the above-mentioned ones: internal pub-sub daemons listen for events and asynchronously applies the relevant actions in near real-time.

3 Challenges and Experiences

A selection of experiences and challenges in developing the big-data marketplace application are discussed in the following. Due to space constraints, a main issue is discussed first, and additional issues are briefly described in Section 3.1.

A main challenge in managing the whole marketplace is the heterogeneous data model of different objects (offers, products, user profiles, photos, comments, etc.). Such items must be promptly presented to the clients and their updates (transactions, see Section 2.2) generate news to be efficiently dispatched to the users.

While the NoSQL [4] choice (see Section 3.1) allows deeply heterogeneous data to be stored together, the presentation and dispatching phases do not friendly handle such diversity. Hence, we introduced an intermediate phase, where almost-homogeneous documents (named entries) are generated from the original objects, and eventually an ad-hoc inverse-indexed dispatching phase prepares the news inbox for each user.

The complete data model consists of the following collections².

- **Original objects.** All first-class objects, such as offers, products, and profiles, have their own collections, thus allowing more efficient indexing and retrieval. All object operations are applied directly on these items.
- **Entries.** Every object has a corresponding entry document in the entries collection. Its purpose is to provide an homogeneous summarized representation across heterogeneous objects to be exploited when presenting search results, news feeds, and update notifications. For instance, each entry has

² In NoSQL, usually *collections* correspond to the relational database tables, and *documents* correspond to records.

title, summary, and icon fields, besides some additional metadata such as the list of shops it refers to. A direct link to the original object is stored to allow immediate retrieval of the full content when needed, whereas the original document has a reference to its entries for reverse lookup. When the original object is inserted, changed, or removed, the corresponding entries are created, updated, or deleted. The change in the entries is applied asynchronously by an event-driven process to avoid locks and ease scaling, in an eventually consistent fashion. Inconsistencies are addressed by forcing the few most critical operations to refer back to the original item (e.g., booking an offer), while the most common and less critical operations can be optimistically handled on the entries themselves (e.g., showing results).

- **Morcels.** Entries are optimized to present contents. They are particularly useful for searching different contents on the marketplace. Instead, when a user goes online and wants to know her updates, e.g., new offers of her shops, new events from her friends, nearby promotions, etc., scanning all entries for relevant content would be unfeasible. Hence we devised an ad-hoc collection of morcels³, tiny documents that simply points to the relevant entries from the end-user point of view. Morcels are an inverse-indexed entry-pointers collection, where each user finds her own updates ready to be dispatched.

3.1 Additional Issues

The design and development of the big-data marketplace application involved addressing many different issues. Some additional key choices are briefly discussed in the following.

Development architecture. From a developer point of view, the platform consists of some Twisted Matrix projects [5] written in Python [6] and provides four abstraction layers: (i) a front-end layer, which is device specific (tablet, browser, mobile device) and handles data presentation; (ii) an API layer, which responds to requests from the front-end, performs authentication, authorization and sanitizes parameter data; (iii) a manager layer, which knows how to actually perform operations on the objects (profiles, offers, etc.); and (iv) a back-end layer, which knows where to retrieve and store data by connecting to proper DBs. The first one runs on clients, whereas the remaining layers are distributed on different servers to allow scaling computational resources.

Data storage. The big-data application is currently running on a cluster of MongoDBs [7]. The choice of a NoSQL database was easily motivated by a flexible data model: lists of values (e.g., tags) and associative arrays with variable length and content are handled as native types.

We initially used Apache CouchDB [8], then evaluated a few alternatives (Cassandra) [9], but some features of MongoDB made it a better choice in our context: native geospatial indexing to find content (shops, offers) by proximity, built-in distributed database handling to provide replication and sharding, which is critical to scale horizontally, and finally a rich query language [7].

³ From *morcel* or *morsel*, a small fragment or share of something, commonly applied to food. Source: <http://en.wiktionary.org/wiki/morsel>

Replication. The back-end databases are configured in replica sets, as in MongoDB definition. Each replica set has a master primary database and one or more secondary slaves. Only the master node can accept write operations, and if it fails or becomes inaccessible, the slaves can autonomously elect a new master. Read operations are sent to the master as default, however, to lighten the master node, they are redirected to slaves on a per-connection basis when data freshness is not strictly required. Replication allows scaling read operations, besides providing fault tolerance and redundancy, however to scale write operations sharding is required.

Sharding. We have configured sharding at collection level (i.e., for each database table separately). Its most critical choice is the sharding key, which must be a field present in each document (i.e., record). Bad choices of the sharding key are typically those involving timestamps or monotonically increasing identifiers⁴. In our application, transactions are heterogeneous (only a small subset of the fields are present in all documents) and the only candidate sharding keys are the source and destination fields, indicating the entities participating in the transaction. The destination entity proved to be a better sharding key. Other sharded collections are entries and morcels. Due to space constraints, we cannot further elaborate on these choices.

Group statistics. Grouping operations are performed on the transactions to compute totals over specific dimensions, such as per customer, per shop and over time. To this aim, we designed an incremental MapReduce [10]: each time it is run, it starts from the last mapreduced transaction and only aggregates new values from the subsequent transactions, hence processing only a minimal subset of data. When a strict real-time updated value is required (e.g., limited offer availability), a query on the latest non-mapreduced transactions and an optimistic approach (e.g., try and check later) have been preferred.

An additional challenge in mapreducing is represented by the validity periods. Among the additional transaction fields, validity periods represent a key feature for bonuses, offers, and products which expires at a given date or in a time frame. The group operation must be able to easily compute both the current activation transition (publishing an offer, assigning a bonus), and the future deactivation or expiration (offer expired, bonus not valid anymore). Both transactions are generated when a new object is created. To this aim, we designed transactions that are inserted in the present but have effect in the future, by means of a special field indicating the operation validity date. As a consequence, the MapReduce has been structured to aggregate such values only at the right time.

4 Conclusions

We presented challenges and design choices of a big-data marketplace application running in the cloud. The marketplace exploits a physical presence in the stores to bridge the gap between the online e-commerce world and the offline physical places and shops. It aims at engaging all actors, customers and merchants, both

⁴ Hashed sharding recently released with MongoDB 2.4 has not been considered yet.

virtually and in the real-world, and managing the wealth of information generated by their shopping-related activities: the platform stores, processes, and extracts actionable knowledge from the huge amount of information of online and offline business activities and their customers.

The big-data application presented in the paper is live as an innovative marketplace service fully forged by a Turin-based start-up in Italy, and it is currently in an early public stage under the commercial name of Desidoo.

As the platform grows, more and more challenges will have to be faced. In the mid term, future works will address the application of the Hadoop framework for batch statistics and possibly for real-time analytics, the redesign of the transactions to model more complex interactions, and the exploitation of solid relational databases for homogeneous subsets of data. In the long term, the marketplace shows promising flexibility to be converted to a Platform as a Service (PaaS) able to provide different application services to the current and new external actors.

References

1. Pallis, G.: Cloud computing: the new frontier of internet computing. *IEEE Internet Computing* 14(5), 70–73 (2010)
2. Murphy, M., Meeker, M.: Top mobile internet trends. KPCB Relationship Capital (2011)
3. Desidoo, by Ooros srl, Turin, Italy, <http://www.desidoo.com/> (last access on May 2013)
4. Stonebraker, M.: Sql databases v. nosql databases. *Communications of the ACM* 53(4), 10–11 (2010)
5. Fettig, A.: *Twisted network programming essentials*. O'Reilly Media, Incorporated (2005)
6. van Rossum, G., et al.: Python programming language website (2007), <http://www.python.org>
7. Chodorow, K., Dirolf, M.: *MongoDB: the definitive guide*. O'Reilly Media (2010)
8. Anderson, C.: *Apache couchdb: The definitive guide*, <http://couchdb.apache.org/>
9. Hewitt, E.: *Cassandra: the definitive guide*. O'Reilly Media (2010)
10. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. *Communications of the ACM* 51(1), 107–113 (2008)

GraphDB – Storing Large Graphs on Secondary Memory^{*}

Lucas Fonseca Navarro¹, Ana Paula Appel²,
and Estevam Rafael Hruschka Junior¹

¹ Universidade Federal de São Carlos
lukinhafn@gmail.com, estevam@dc.ufscar.br

² IBM Research Brazil
apappel@br.ibm.com

Abstract. The volume of complex network data has been exponentially increased in the last years madding graph mining area the focus of a lot of research efforts. Most algorithms for mining this kind of data assume, however, that the complex network fits in primary memory. Unfortunately, such assumption is not always true. Even considering that, in some cases, using big computer clusters (in a MapReduce fashion, for instance) might be a suitable way to circumvent part of the difficulties of mining big data, efficiently storing and retrieving complex network data is still a great challenge. Thus the main goal of this work is to introduce the definition of a new data structure, called *GraphDB-tree* that can be used to efficiently store and retrieve complex networks, and also, allowing efficient queries in large complex networks.

1 Introduction

Over the past years the amount of collected and stored data has been substantially increased and the World Wide Web is the one of the main actors in this scenario. The large volume of available data, the low cost of storage, the stunning success of online social networks and web2.0 applications all lead to complex network of unprecedented size. Typical graph mining algorithms assume that data (from complex networks in this case) fit in the memory of a typical workstation; however there are real complex networks that violate this assumption, spanning multiple Giga-bytes, and heading to Tera and Peta-bytes of data.

MapReduce is a programming framework 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 academia. However, even with the availability of distributed computational resources through the Cloud, choosing the best way to partition a complex network for distributed computation is still a challenge. Also, for programmers, debugging and optimizing distributed algorithms are still a difficult and expensive task [1].

^{*} The authors thank Carnegie Mellon University, CNPq, FAPESP and Capes.

Index structures have been efficiently used to handle relational database systems (RDBMS). Such structures which were mainly used to handle only numbers and small text, now are being required to handle complex data such as images, video, DNA sequence and so on [2]. Indexing approaches are responsible for the efficiency of RDBMS queries, and also, they are used in several data mining algorithms such as clustering. Also, RDBMS are widely used in thousands of companies and this will not change, meaning companies will not through out their databases to use a completely different system that will not suit well with their applications. Online systems not only produce unstructured data but a lot of transactional data that can be mapped as a complex network (join operation). Considering the aforementioned scenario, some important questions arise. Would be possible to efficiently handle large complex network in one single ordinary desktop machine? Would be feasible to build a simple index structure able to support graphs operations?

In this paper we explore possible answers to the previous questions. Also, we present *GraphDB-tree*, a simple index structure used to store and query large complex networks and to support data mining algorithms. The main motivation for proposing *GraphDB-tree* is to store large networks at *Centaurs* [3]. *Centaurs* is a framework to mine large complex networks that is used as a component of NELL [4]. The framework combines graph mining algorithms, specially those related to community detection and link prediction, to find missing edges that were lost during the building process of the *Read the Web*¹ graph [5].

Experiments with *GraphDB-tree* were performed using an ordinary desktop computer and the obtained results are up to 70% more efficient than *Graph-tree* proposed in [6].

The sequence of 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 experiments and results and Section 6 concludes the work.

2 Definitions

A graph is a useful way to specifying relationships among a collection of items. It consists of a set of objects, called nodes, with certain pairs of these objects connected by links called edges. Two nodes are neighbours if they are connected by an edge. A complex network can be modeled as a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, in which \mathcal{V} represents the number of nodes/vertex, and \mathcal{E} represents the number of edges/links. The traditional way of computationally representing a graph \mathcal{G} is based on the adjacency matrix, which is a square matrix $\mathbf{A} = N \times N$, with $N = |\mathcal{V}|$, and $\mathbf{A}_{i,j} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and 0 otherwise.

A graph is *undirected* 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* as a set of nodes (same as in the undirected case) together

¹ <http://rtw.ml.cmu.edu/rtw/>

with a set of directed edges; each *directed edge* is a link from one node to another, with the direction being important. *Node degree*, also called *neighbourhood*, is defined by the amount of incident edges. Another important concept is related to the *triangles* that are triples of fully connected nodes.

A triangle $\Delta(G)$ of a graph $G = (V, E)$ is a three-node sub-graph with $V_\Delta = \{u, v, w\} \in V$ and $E_\Delta = \{(u, v), (v, w), (w, u)\} \in E$. An open triangle $\Lambda(G)$ of a graph $G = (V, E)$ is a three node sub-graph where $E_\Lambda = \{(u, v), (v, w)\} \in E \wedge \{u, w\} \notin E$. The transitivity ratio is the fraction of closed triangles in the network, that is, $T(G) = \frac{3 * \Delta(G)}{\Lambda(G)}$.

3 Related Work

In the last years there has been a significant increase in the volume of available unstructured data, specially fueled by complex networks data available on the Web. on the other hand, investigation and research on how to efficiently store complex networks in secondary memory has gain almost no attention. In this area, the majority of studies focus on indexing databases of small graphs, where the main task is to search for similar graphs or sub-graphs [7].

Hadoop is the open source implementation of MapReduce [8], which provides a Distributed File System (HDFS) and PIG, a high level language for data analysis [9]. Based on Hadoop, there is a number of graph mining packages for handling graphs with billions of nodes and edges such as PeGaSus [10].

A interesting comparison between parallel DBMS and MapReduce is presented in [11]. As the author says, 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 needed for MapReduce architecture to be highly adopted.

There are other examples of NoSQL architectures. A NoSQL graph database that has attracted a lot of attention is Neo4j, which is an open source graph databases. According to Neo4j² website, Neo4j is "an 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), it has an API that is easy to use, and supports efficient traversals. However, most of the algorithms used in this database is base on paths algorithms like Dijkstra, A* and so on, thus, are not very useful for graph mining, mainly because of their high computational complexity. A comparative study of Neo4j with a relational database is presented in [12], where the authors show that Neo4j is not ready for graph storing, not only for the type of queries but also by other properties as multi-user and security.

² <http://neo4j.org>

Another interesting approach in graph management is related to RDF (Resource Description Framework) data, which 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 representing 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 [13] [14]. 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

An adjacency matrix is a convenient graph representation in many cases because most of calculations can be easily done using such structure. However, not all complex network applications are suitable to be represented by an adjacency matrix. For example, to recover all the neighbors of a node it is necessary to scan the corresponding row in the adjacency matrix and search for non-zeros. This search takes time $O(n)$, since it is the length of row and in a network. Also, for a large n , it could mean a lot of time. Most of complex networks are sparse with most of the nodes been one degree node, which makes an adjacency matrix inefficient. But if it is possible to have an adjacency matrix in which the search for a node was $O(T(n))$ and it is not memory consuming, it would be a nice solution.

The adjacency list is the most widely computational representation used for complex networks. An adjacency list can store networks with multi-edges or self-edges. Thus, in this work we developed an adjacency list representation that works in secondary memory. The main advantage in using this structure is that the complex network can be represented in a very compact way and it allows fast access, which is a great benefit for secondary memory data structures. An example of efficient use of adjacency list is METIS algorithm, however its implementation is based in main memory [15].

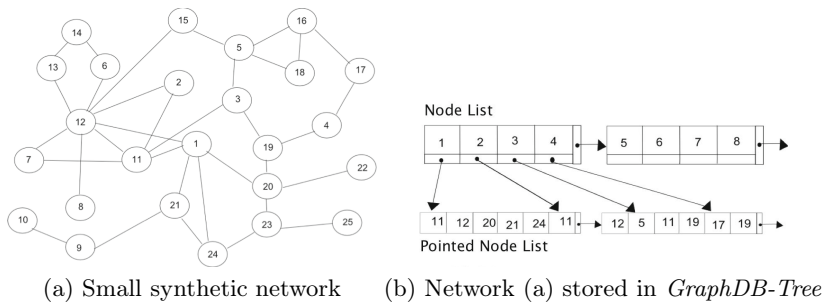


Fig. 1. *GraphDB-Tree* example

A secondary memory adjacency list requires, however, a carefully implementation, since its compactness can be lost if the nodes do not share disk pages. Because of the sparsity of most complex networks and very frequent one-degree nodes, if one disk page is allocate to nodes with small degrees (degree less than the number of nodes that fits in one disk page) the structure is wasting a lot of disk space. Thus, a naive implementation can make an adjacency list expensive data structure.

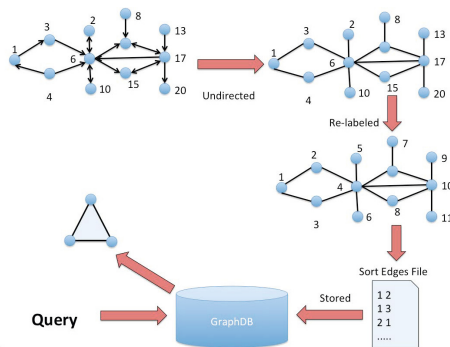


Fig. 2. *GraphDB-Tree*

GraphDB-Tree reads an edge list $\langle v_i, v_j \rangle$ from a complex network G and stores each distinct v_i on a disk page called node list, presented in Figure 4 (b). The target nodes v_j are pointed for each v_i and stored in continuous disk pages called target node list pointed to nodes v_j .

Each node v_i in the list node has stored the node identification (id), its degree, a pointer to the target node list and the position where its target list start on the target node list. Storing the node degree helps to know how many nodes are necessary to read whenever it is necessary to recover all neighbors of node v_i . Each target node list page stores target ids and a pointer to the next page. If we want to store a graph with weighted or labeled edges, we can simple change how to store a node in the target node list. Also, *GraphDB-tree* supports directed and undirected networks.

Figure 4 presents an example of a network (a) and in (b) a simple representation of how it is stored in *GraphDB-tree*. For example, node $v_i = 1$ has a neighborhood $v_j \in \{11, 12, 20, 21, 24\}$, so in the node list we have all nodes *ids* from the complex network (a), for node $v_i = 1$ we have a pointer to the target list page where all its neighbors are stored, the neighbors are stored sorted.

The first task in the proposed approach is to efficiently store a network. Doing this, a traditional graph mining algorithms can be applied even when the network representation is too big to be fully stored in main memory all at once. In Figure 2 we present how *GraphDB-Tree* works. It copes with directed and undirected networks, but a directed network requires some extra computation to convert it

Algorithm 1. Counting All Triangles in *GraphDB-tree***Require:** G **Ensure:** Number of Triangles

```

1: for  $count \leftarrow 1$  to  $VpageNumber$  do
2:   Load page  $\leftarrow count$ 
3:   for  $i \leftarrow 0$  to  $VpageCapacity$  do
4:     AdjVector  $\leftarrow$  Adjacency list from node  $i$ 
5:     for  $j \leftarrow 1$  to  $adjNum$  do
6:       if  $pageID(AdjVector[j]) > count$  and  $(AdjVector[j] > id(i))$  then
7:         Access edge list page of  $AdjVector[j]$ 
8:         for  $k \leftarrow 1$  to  $degree(AdjVector[j])$  do
9:           for  $l = j + 1$  to  $adjNum$  do
10:            if  $AdjVector[l] > id(i)$  then
11:              if  $AdjVector[l] == k - th$  element of adj list of  $AdjVector[j]$ 
12:                then
13:                   $CloseTriangle ++$ 
14:                else
15:                   $OpenTriangle ++$ 
16:                end if
17:              end if
18:            end for
19:          end if
20:        end for
21:      end for
22:    end for

```

to an undirected one. After that, the network nodes ids should be relabeled in case ids are not sequential. Edges file should also be sorted.

There is a large number of queries that might be interesting to implement for testing the efficiency of a new network data structure, such as page rank, two hops, triangles and so on. Plenty of researchers have investigated the behavior of triangles on a network and how they can be used to indicate the existence of larger cliques. Cluster coefficient measures the "cliqueness" degree of a graph. Thus, one of the operations of interest is the estimation of the clustering coefficient and the transitivity ratio, which respectively translates to the number of triangles in the network, or the number of triangles that a node participates in [16].

Considering that the main focus of *GraphDB-Tree* is to support link prediction algorithms to be used in NELL, query all triangles was the first task implemented, since triangles are the base for link prediction. Using a traditional RDBMS to query all triangles is expensive since it is a tree-way join. Algorithm 1 presents the triangle counting procedure implemented in *GraphDB-Tree*. Our algorithm sequentially passes through each node, because of that, it passes in each $VpageNumber$ (1), and then in each node from the page (3), thus, the representation can be done as in `page[cont].node[i]`. The adjacency list of each

node[i] is extracted and stored in an auxiliary vector (4), self-loops are removed during this process. Then, each position of the auxiliary vector is visited, loading its pages if needed (some node pages can be already loaded), and then, each node from his adjacency list (8) is visited, trying to find whether this node is in the adjacency vector of node[i] (11). If it is, then we got a close triangle, whereas if it is not, we got at least an open one. Conditions (6) and (10), are used to ensure each triangle in the graph is counted only once. In this sense, the algorithm does not need to count all triangles and divide the result by 3 (as done in many other approaches). As our experiments show in Section 5, *GraphDB-Tree* is efficient for store and retrieve all triangles networks with billions of edges.

5 Experiments and Results

In this section we present the results achieved by *GraphDB-Tree* in specific scenarios. The experiments were performed in a computer with Intel(R) Core(TM) i7 2.40GHz with 6 Giga bytes of RAM and running Linux 11.10 (32bits) and we used 14 real networks from SNAP Website³. Table 1 presents a description of each network.

Table 1. Real datasets description and query time. Respectively the number of nodes ($|V|$), edges ($|E|$), triangles ($|\Delta|$), insertion time in *GraphDB-Tree* (I), time to query all triangles (Δ), the transitivity ratio in each network ($T(G)$), Size in MB to store networks using *GraphDB-Tree* and time to query all triangles in R

name	$ V $	$ E $	$ \Delta $	I	Δ	$T(G)$	size	R
ca-GrQc	5,242	28,980	48,260	1	1	0.6298	0.47	1
wiki-Vote	7,115	201,525	608,389	1	8	0.1255	1.78	22
Ca-HepPh	12,007	237,001	3358499	1	7	0.1457	4.21	18
Cit-HepTh	27,770	704,610	1,478,735	1	14	0.1196	6.38	60
Email-EuAll	265,214	730,052	267,313	1	36	0.0041	12.3	925
RoadNet-ca	1,965,206	5,533,214	120,676	3	9	0.0604	92.1	12
Web-google	875,713	8,643,937	13,391,655	3	83	0.0552	90.7	7021
WikiTalk	2,394,385	9,319,131	9,203,519	5	7,523	0.0011	132	43200
As-skitter	1,696,415	22,190,495	28,769,868	9	7,523	0.0054	219.5	+21600
Cit-Patents	3,774,768	33,037,896	7,515,023	15	121	0.0671	357	308
soc-Pokec	1,632,803	44,603,930	32,557,458	28	68,411	0.0161	398.2	9271
Com-LiveJournal	3,997,962	69,362,379	177,820,130	39	3,410	0.1154	654.8	19740
Soc-LiveJournal	4,847,570	86,054,328	285,030,584	42	13,382	0.2882	809.1	overflow
Com-Orkut	3,072,441	234,370,167	633,319,568	112	80,492	0.2303	1974.4	overflow

To show *GraphDB-Tree* efficiency we present the results for the task of querying for all triangles. Since triangles are intrinsic to undirected networks, each directed network was preprocessed by removing the direction. Table 1 presents

³ <http://snap.stanford.edu/>

respectively the number of nodes ($|V|$), edges ($|E|$), triangles ($|\Delta|$), insertion time in *GraphDB-Tree* (I), time to query all triangles (Δ), the transitivity ratio in each network ($T(G)$) and the time using software R to counting all triangles. Time is given in seconds in all columns and represents the average of running the algorithm three times. We also compare *GraphDB-Tree* results with a traditional triangle counting implemented in the R⁴ software. However, for some of the networks it was not possible to run R triangle counting as we got memory overflow. For one network it took more than 6 hours and we aborted the process after that. Another interesting observation is that R takes much more time to load the network in memory than *GraphDB-Tree*

Table 2 presents the number of nodes, edges, triangles, number of accessed disk pages and query time in seconds of several synthetic networks generated following the Small World model to perform a scalability experiment. All the networks were generated with the rewiring probability equal to 0.5.

Table 2. Small World Networks Description

# Nodes	# Edges	# Triangles	# Disk Access	Query Time
100,000	1,000,000	565,780	1,238,590	5
100,000	5,000,000	15,434,811	6,599,634	134
100,000	10,000,000	62,856,260	15,459,481	1,218
1,000,000	10,000,000	5,631,498	12,442,169	85
1,000,000	50,000,000	153,334,892	66,700,095	1,387
5,000,000	50,000,000	28,122,113	62,239,827	296
2,500,000	62,500,000	93,776,741	76,817,011	814
1,000,000	100,000,000	619,875,841	157,668,566	9,727

In Figure 5 (a) we present the scalability to query all triangles. We can see that WikiTalk, Web-Google and RoadNet-ca represented by region A have almost the same number of edges, however they present a very different time to perform the query. However, if we observe, in Table 1 the transitivity ratio of WikiTalk is the smallest one while the time to query all triangles is the highest one. In region B we have WikiTalk and As-skitter, both with transitivity ratio very low but with different number of edges. This means that *GraphDB-Tree* is more efficiently for clustered networks. Of course, the number of edges matters, since for large graphs the query time will be high anyway.

In Figure 5 (b) we plot time against the number of edges and as we can see the execution time has a tendency to be super linear, which makes *GraphDB-Tree* a suitable structure to store and handle large complex networks. Even the fit is not with a high correlation coefficient, we want to show that the behavior of synthetic networks were the same of real networks, with the same happening with networks with same number of edges but more clustered. We did not run *Graph-Tree* proposed in [6], but looking for the experiments with same networks

⁴ <http://www.r-project.org/>

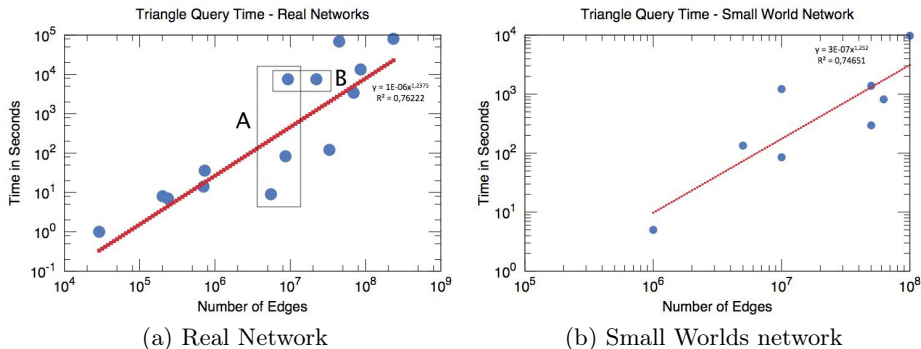


Fig. 3. Scalability test

and similar hardware *GraphDB-Tree* is up to 70% faster. For example, *Graph-tree* takes 8,579 seconds to counting all triangles while *GraphDB-Tree* takes 121 seconds.

6 Conclusion and Future Work

The main contribution of this work is the proposition of a new data structure on secondary memory to store large complex networks called *GraphDB-Tree*. We developed *GraphDB-Tree* having in mind that it must be generic enough to allow other networks and other graph mining tasks (as community detection and link prediction, for instance) to take advantage of the proposed approach. Thus, *GraphDB-Tree* is a versatile data structure that allows not only undirected graphs as well as directed, weighted and labeled networks. As the experiments have empirically shown, *GraphDB-Tree* is scalable and not time consuming. *GraphDB-Tree* easily stored networks with millions of edges and is up to 70% faster than other methods based on the same kind of data structures. There are other algorithms that can be supported by *GraphDB-Tree* such as page rank, two hops forward and backward, METIS for graph partition and so on. However the most simple one is link prediction as common neighbors and Adamic/Adar since they are based on open triangles that is a result obtained for free in our structure (when we count all triangles, as we can see in Algorithm 1). Also, for now the structure is being used only for static networks, since most of the counting we are doing is one way counting. However to support NELL we plan to extend the structure, for example using *B-Tree* polices occupying only 50% of a node. This police could be adapt for each node has a space in the end of node list or new nodes could be added among the edges list. Another extension that in test phase is use cache memory methods to maintain in memory the most common pages.

References

1. Kyrola, A., Blelloch, G., Guestrin, C.: Graphchi: large-scale graph computation on just a pc. In: Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation, OSDI 2012, pp. 31–46. USENIX Association, Berkeley (2012)
2. Traina Jr., C., Traina, A.J.M., Seeger, B., Faloutsos, C.: Slim-trees: High performance metric trees minimizing overlap between nodes. In: Zaniolo, C., Grust, T., Scholl, M.H., Lockemann, P.C. (eds.) EDBT 2000. LNCS, vol. 1777, pp. 51–65. Springer, Heidelberg (2000)
3. Appel, A.P., Hruschka Jr., E.R.: Centaurs a component based framework to mine large graphs. In: XXV Brazilian Symposium on Databases, Belo Horizonte, MG, Brazil, pp. 1–8 (2010)
4. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence, AAAI 2010 (2010)
5. Appel, A.P., Hruschka Jr., E.R.: Prophet - a link-predictor to learn new rules on nll. In: 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW), Vancouver, BC, Canada, December 11, pp. 917–924 (2011)
6. Pereira, A.L., Appel, A.P.: Modeling and storing complex network with *graph-tree*. In: New Trends in Databases and Information Systems, Workshop Proceedings of the 16th East European Conference, ADBIS 2012, Pozna, Poland, September 17-21, pp. 305–315 (2012)
7. Angles, R., Gutierrez, C.: Survey of graph database models. *ACM Comput. Surv.* 40, 1:1–1:39 (2008)
8. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. *Commun. ACM* 51(1), 107–113 (2008)
9. 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)
10. 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, Columbus, Ohio, April 29- May 1, pp. 548–558 (2010)
11. 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: SIGMOD Conference, pp. 165–178. ACM (2009)
12. 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)
13. Weiss, C., Karras, P., Bernstein, A.: Hexastore: sextuple indexing for semantic web data management. *Proc. VLDB Endow.* 1(1), 1008–1019 (2008)
14. 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)
15. Karypis, G., Kumar, V.: Parallel multilevel k-way partitioning for irregular graphs. *SIAM Review* 41(2), 278–300 (1999)
16. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. *Nature* 393(6684), 440–442 (1998)

Hadoop on a Low-Budget General Purpose HPC Cluster in Academia

Paolo Garza, Paolo Margara, Nicolò Nepote, Luigi Grimaudo, and Elio Piccolo

Dipartimento di Automatica e Informatica, Politecnico di Torino,
Corso Duca degli Abruzzi, 24 10129 Torino, Italy
`name.surname@polito.it`

Abstract. In the last decade, we witnessed an increasing interest in High Performance Computing (HPC) infrastructures, which play an important role in both academic and industrial research projects. At the same time, due to the increasing amount of available data, we also witnessed the introduction of new frameworks and applications based on the MapReduce paradigm (e.g., Hadoop). Traditional HPC systems are usually designed for CPU- and memory-intensive applications. However, the use of already available HPC infrastructures for data-intensive applications is an interesting topic, in particular in academia where the budget is usually limited and the same cluster is used by many researchers with different requirements. In this paper, we investigate the integration of Hadoop, and its performance, in an already existing low-budget general purpose HPC cluster characterized by heterogeneous nodes and a low amount of secondary memory per node.

Keywords: HPC, Hadoop, MapReduce, MPI applications.

1 Introduction

The amount of available data increases every day. This huge amount of data is a resource that, if properly exploited, provides useful knowledge. However, to be able to extract useful knowledge from it, efficient and powerful systems are needed. One possible solution to the introduced problem consists in adopting the Hadoop framework [6], which exploits the MapReduce [1] paradigm for the efficient implementation of data-intensive distributed applications.

The recent years have also witnessed the increasing availability of general purpose HPC systems [3], such as clusters, commonly installed in many computing centers. They are usually used to provide different services to communities of users (e.g., academic researches) with different requirements. These systems are usually designed for CPU- and memory-intensive applications. However, we witnessed some attempts to integrate Hadoop also in general purpose HPC systems, in particular in academia. Due to limited budgets, the integration of Hadoop in already available HPC systems is an interesting and fascinating problem. It will allow academic researches to continue to use their current MPI-based applications and, at the same time, to exploit Hadoop to address new (data-intensive) problems without further costs.

In this paper, we describe the integration of Hadoop inside an academic HPC cluster located at the Politecnico di Torino computing center “HPC@polito”. This cluster, called CASPER, hosts dozens of scientific softwares, often based on MPI. We decided to integrate Hadoop in CASPER to understand if it is able to manage also huge data or if an upgrade is needed for this new purpose. Our main goals consist in continuing to provide the already available services, based on MPI applications, and the new ones based on Hadoop using the same system.

2 HPC@polito

HPC@POLITO (www.hpc.polito.it) aims at providing computational resources and technical support to both academic research and university teaching. To pursue these goals, the computing center has set up a heterogeneous cluster called CASPER (Cluster Appliance for Scientific Parallel Execution and Rendering) with a peak performance of 1.2 TFLOPS. The initiative counts 25 hosted projects and 12 papers developed thanks to our HPC and published by groups operating in different research areas.

A detailed technical description of the system is available in previous papers [2] [5]. In our vision CASPER is a continuously evolving system, developed in collaboration with several research groups, and being renewed regularly.

2.1 Cluster Configuration

CASPER is a standard MIMD distributed shared memory InfiniBand heterogeneous cluster. It has 1 master node and 9 computational nodes, 136 cores, and 632 GB of main memory. From Figure 1, you can see that the system is composed of three different types of computational nodes, which have been added to the cluster in subsequent stages according to the needs of the research groups. The cluster is evolving into a massively parallel, large memory system, having average cpu speed and many cores per node. The nodes have very small and low performance local hard disks, which have been designed to contain only the operating system; experimental data are maintained in a central NAS.

CASPER is normally used through the scheduler/resource manager SGE (now OGE) for running custom code or third-party software, often taking advantage of MPI libraries (e.g., *Esys-Particle*, *Matlab*, *OpenFOAM*, *Quantum-Espresso*). The cluster configuration is therefore a compromise that provides sufficient performance for all of the cited softwares. However, the current applications are rarely data-intensive and do not use huge data.

2.2 Hadoop Deployment

The installation of Apache Hadoop was made trying to harmonize its needs to those of our specific system. CASPER is an installation of Rocks Cluster 5.4.3¹.

¹ <http://www.rocksclusters.org>

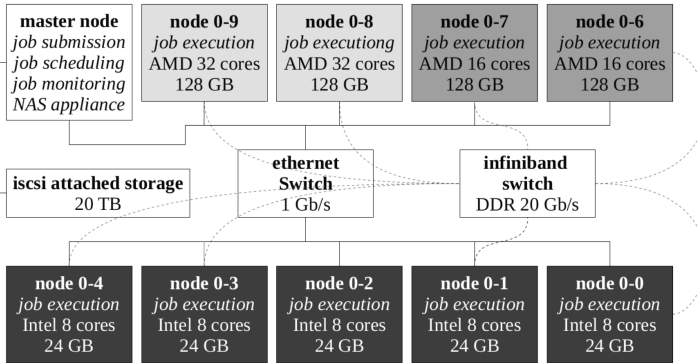


Fig. 1. CASPER cluster configuration, early 2013

To install Hadoop 1.0.4 on it, we used the packages from the EPEL repository for CentOS, from which the Rocks Cluster distribution derives.

We decided to use the master node of the cluster also as master node of Hadoop, while the computation nodes were configured as slave nodes. HDFS was configured with permissions disabled, data block size set to 256MB and number of replicas per data block set to 2. On five nodes (nodes from 0-0 to 0-4), we added 1TB local hard disks for the exclusive use of HDFS. On the remaining four nodes, data was stored in `/state/partition1`, which is a partition created by the Rocks node installer and corresponds to all the remaining space on the local disk (the local disk size is 160GB for these four nodes). We configure Hadoop by taking into consideration the heterogeneous nature of CASPER. More specifically, for each node the maximum number of mappers was set equal to the number of cores, while the maximum number of reducers was set to 4 for the five nodes with a large and efficient 1TB local disk, and to 0 for the other four nodes due to the slow efficiency and small size and their hard disks.

The package provided by Oracle to integrate Hadoop into our version of SGE is incompatible with the current version of Hadoop. Hence, we decided to use an integration approach based on the creation of a dedicated queue and a new Parallel Environment in the SGE scheduler, so that the Hadoop tasks are still subjected to Hadoop but managed through the queuing system of the cluster.

3 Experimental Results

As described in the introduction of the paper, for some future research applications we need to use CASPER on large/huge data. However, on the one hand we cannot buy a new dedicated cluster. On the other hand we cannot dismiss the softwares already hosted on CASPER. Hence, we performed an initial set of experiments to understand the scalability, in terms of data size, of an MPI-application. These experiments allowed us to identify the data size limit of MPI programs on our cluster. Then, we performed a set of experiments based on

Hadoop (i) to evaluate the scalability of Hadoop on CASPER and (ii) to understand which upgrades are needed to be able to use CASPER on big data.

3.1 Experimental Setting

To evaluate the scalability of MPI-based algorithms, we used an MPI implementation of the quicksort algorithm derived from a public available code [4]. It receives in input a single file containing a set of numbers (one number per line) and generates one single file corresponding to the sorted version of the input one. The algorithm, similarly to all typical MPI applications, works exclusively in main memory.

One of the benchmarking test usually performed to analyze efficiency and scalability of Hadoop is the Hadoop-based implementation of Terasort [6]. Hence, we did the same also to test the scalability of Hadoop on our cluster.

3.2 MPI-Based Sorting Algorithm

The scalability of the MPI sorting algorithm was evaluated on files ranging from 21GB to 100GB. Larger files are not manageable due to the memory-based nature of the algorithm. Detailed results are reported in Table 1(a) for three difference configurations, characterized by a different number of nodes/cores. We considered initially only the nodes 0-8 and 0-9 reported in Figure 1, then nodes from 0-6 to 0-9, and finally all nodes. The first configuration is homogeneous but it is characterized by only 64 cores, while the last one is the most heterogeneous but it exploits all the available resources.

The results reported in Table 1(a) highlight that our MPI sorting algorithm is not able to process file larger than 100GB. Hence, it cannot manage big data. As expected, the execution time decreases when the number of nodes/cores increases. Figure 2(a) shows that the execution time decreases linearly with respect to the number of cores. However, the slope of the curves depend on the file size. Hence, the availability of more cores is potentially a positive factor. However, the increase of the number of nodes, in some cases, has a negative impact. More

Table 1. Execution time

(a) MPI-based sorting algorithm.
DNF=Did not finished.

Dataset size (GB)	Configuration #cores/Total RAM(GB)	Execution time
21GB	64 cores/256GB	39m49s
	96 cores/512GB	31m52s
	136 cores/632GB	26m30s
42GB	64 cores/256GB	1h41m32s
	96 cores/512GB	1h24m41s
	136 cores/632GB	DNF
100GB	64 cores/256GB	3h38m59s
	96 cores/512GB	2h52m59s
	136 cores/632GB	DNF

(b) Hadoop-based terasort.

Dataset size (GB)	Configuration #cores/Total RAM(GB)	Execution time
100GB	40 cores/120GB	1h23m2s
	136 cores/632GB	57m26s
200GB	40 cores/120GB	3h52m57s
	136 cores/632GB	3h21m49s

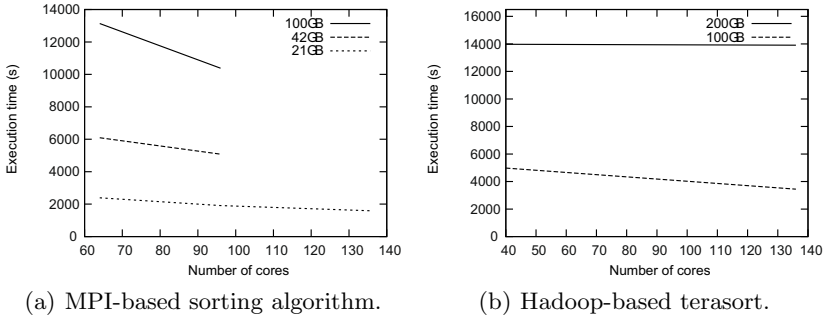


Fig. 2. Execution time

specifically, if we use all nodes, the sorting process does not end with files larger than or equal to 42 GB. The problem is given by the (limited) size of the RAM of the 5 Intel nodes (24GB per node). They are not able to process the tasks assigned to them by the MPI-based sorting program when the file size is larger than approximately 40GB.

3.3 Terasort (Hadoop-Based Application)

Since the MPI application does not allow processing large files on CASPER, we decided to test Hadoop on it. Hadoop is usually used on commodity hardware. However, CASPER has a set of peculiarity (e.g., it is extremely heterogeneous) and hence it could be not adapt for Hadoop. We performed the tests by means of a standard algorithm that is called Terasort. We decided to evaluate Hadoop-based implementation of Terasort on two extreme configurations. The first configuration is based only on the 5 Intel nodes (40 cores), while the second one exploits all nodes (136 cores). The first configuration is homogeneous (5 Intel 3.2GHz nodes with 1TB of secondary memory per node). The second one is extremely heterogeneous (different CPU frequencies and local disks with size ranging from 160GB to 1TB).

The results reported in Table 1(b) and Figure 2(b) show that the Hadoop-based Terasort algorithm can process in less than 4 hours a 200GB file. Hence, also on CASPER, which is not designed for Hadoop, the use of Hadoop allows processing files larger than those manageable by means of MPI algorithms. However, the time of the first configuration (5 homogeneous nodes) is slightly slower than the second one (composed of all nodes). The second configuration has +240% more cores than the first one but the execution time decrease is only -13% when the file size is 200GB (-31% when the file size is 100GB). These results confirm that we need more homogeneous nodes in our cluster and on the average larger local disks on each computational node if we want to increase the scalability of CASPER for data-intensive applications based on Hadoop. We will consider this important point during the planning of the next upgrade of CASPER.

General Considerations. Based on the achieved results, we can conclude that CASPER can potentially be used to run both the already hosted MPI-based applications and new Hadoop-based applications. However, some upgrades are needed in order to improve the performance of CASPER on large datasets.

The results reported in Sections 3.2 and 3.3 can be exploited also to decide how to allocate the different applications on CASPER. Hadoop seems to achieve better results when homogeneous nodes, with large and efficient local disks, are used (i.e., the 5 Intel nodes in our current system), while the MPI-based application, which is a main memory-intensive application, seems to perform better on nodes with more efficient processors and a large amount of main memory (i.e., the AMD nodes in our current system). On CASPER, analogously to all traditional clusters, a set of queues can be created. Each queue is associated with a set of nodes and can be characterized by a priority level. Based on the discussed results, the association of the applications based on Hadoop to a queue that includes the 5 Intel nodes, and the association of the MPI-based applications to a queue that includes the AMD nodes seems to be, potentially, a good configuration. This configuration should allow to execute contemporaneously Hadoop- and MPI-based applications.

4 Conclusions

Due to the increasing request of data-intensive applications, we decided to analyze the potentiality of our low-budget general purpose cluster for this type of applications. In this paper, we reported the results of this experience. The performed experiments highlighted the limitations of our current cluster and helped us to identify potential upgrades that should be considered in the future. Further experiments will be performed on other algorithms (e.g., the merge sort algorithm) to confirm the achieved results.

References

1. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. *Commun. ACM* 51(1), 107–113 (2008)
2. Della Croce, F., Piccolo, E., Nepote, N.: A terascale, cost-effective open solution for academic computing: early experience of the dauin hpc initiative. In: AICA 2011, pp. 1–9 (2011)
3. Dongarra, J.: Trends in high performance computing: a historical overview and examination of future developments. *IEEE Circuits and Devices Magazine* 22(1), 22–27 (2006)
4. Maier, P.: qsort.c (2010), <http://www.macs.hw.ac.uk/~pm175/F21DP2/src/>
5. Nepote, N., Piccolo, E., Demartini, C., Montuschi, P.: Why and how using HPC in university teaching? a case study at polito. In: DIDAMATICA 2013, pp. 1019–1028 (2013)
6. White, T.: Hadoop: The Definitive Guide, 1st edn. O’Reilly Media, Inc. (2009)

Discovering Contextual Association Rules in Relational Databases*

Elisa Quintarelli and Emanuele Rabosio

Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria
{elisa.quintarelli, emanuele.rabosio}@polimi.it

Abstract. Contextual association rules represent co-occurrences between contexts and properties of data, where the context is a set of environmental or user personal features employed to customize an application. Due to their particular structure, these rules can be very tricky to mine, and if the process is not carried out with care, an unmanageable set of not significant rules may be extracted. In this paper we survey two existing algorithms for relational databases and present a novel algorithm that merges the two proposals overcoming their limitations.

1 Introduction

Nowadays we are deluged by information coming from many different sources, spanning from the Web to sensor networks. This Big Data, if not properly filtered, risks to confuse the users instead of being a precious resource. A criterion that has been proposed in the literature to select the appropriate knowledge chunk is the notion of *context* [5]. Context-aware systems acquire and exploit information about the environment and the situation that the user is currently living, in order to shape the behavior of the application on his/her needs; the interest for such systems is growing in both research and industry. In data management, the adaptation capability of a contextual system consists in choosing the relevant information to be provided to the user.

More advanced approaches [14,12] have described techniques to refine context-based data tailoring by employing the so-called *contextual preferences*, representing the tastes of the individual users in the various situations. However, the number of possible contexts may be huge, and overburdening the users with the manual specification of preferences for all such contexts is usually not a viable option. As a first step in the development of a methodology to automatically mine contextual preferences from log data, we study the extraction of *contextual association rules* [3,6,11], that are association rules representing co-occurrences between context and other entities; indeed, such rules mirror the structure of contextual preferences. In more detail, in this paper we focus on the problem of mining contextual association rules from relational databases.

The extraction of such rules from relational databases may be very challenging, since the number of discovered patterns may easily become huge and unmanageable, because of a large portion of not significant rules. This is due to some sources of redundancy

* This research has been partially funded by the European Commission, Programme IDEAS ERC, Project 227977-SMScom and by the Italian project Industria 2015, Program no. MI01 00091 SENSORI.

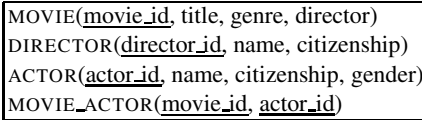


Fig. 1. Database of the running example

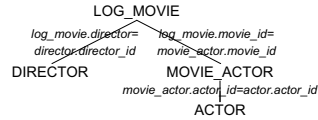


Fig. 2. Join tree of r_1

intrinsic to the relational setting, like primary keys, but also to the special structure of contextual association rules itself that, imposing a context in the antecedent, may lead to mine not well-formed rules.

This paper, to the best of our knowledge, is the first attempt to systematically study the problem of contextual association rule discovery in relational databases. The main contributions towards the solution of this relevant issue are the following: 1) formal definition of the contextual association rules in relational databases, highlighting the challenges connected to their mining (Sect. 2); 2) survey of two existing association rule mining methods – i.e., constrained itemset mining and extraction of frequent conjunctive queries – with respect to their ability to infer contextual rules (Sect. 3); 3) proposal of a novel algorithm joining the advantages brought by the two examined techniques (Sect. 4), and its experimental evaluation (Sect. 5). Finally, Sect. 6 concludes the paper.

The different approaches will be compared using as running example the information system of a company offering services of video on demand, relying on a global relational database. A portion of the schema of the database is shown in Fig. 1.

2 Contextual Association Rules in Relational Databases

In this section we introduce the notions of context and contextual association rules, and the definitions useful for the mining process.

Several context models have been defined in the literature [2,4]. In this paper, to ease the discussion we adopt a simple key-value model envisaging some possible dimensions, representing the perspectives through which the context is analyzed; in our movie scenario the relevant dimensions are: *user*, *interest_topic*, *situation*, *time* and *day*. Each dimension is associated with a fixed set of possible values, and a context can be specified through a conjunction of dimension-value equalities. For example, a valid context is $user = adult \wedge int_topic = cinema \wedge time = night$, which describes the situation of an adult going to the cinema at night.

Definition 1 (Contextual relation). A contextual relation r^C is a relation whose schema $R(X)$ has the attribute set X partitioned into two disjoint subsets: the contextual attributes X_C and the non-contextual attributes X_R .

Note that the contextual attributes correspond to the context dimensions. In our reference scenario, the context represents the situation that the user is living when accessing the data, and so it is *external* to the database; thus, the original database is not composed by contextual relations. However, a contextual relation may be associated with

each relation of the database, as a log of the past data accesses. In the following we will consider a contextualized version of the MOVIE relation, named LOG_MOVIE, storing the history of the movies purchased by several users, also recording the context in which the purchases were performed. The schema of the log relation is as follows:

LOG_MOVIE(id, user, int_topic, situation, time, day, movie_id, title, genre, director)

Definition 2 (Contextual association rule). A contextual association rule on a contextual relation r^C with schema $R(X)$ is a pair $(C \rightarrow cond, T)$ where:

- $C \rightarrow cond$ is an association rule. C is a conjunction of conditions of the form $B = b$, with $B \in X_C$ and b is a context value associated with B . $cond$ is a conjunction of conditions of the form $A = a$, with A belonging to X_R or to the attributes of the relations reachable from r^C with joins on foreign keys; a is a value in the domain of A .
- T is the tree with labeled edges representing the join executed in the consequent of the rule. The tree is rooted in $R(X)$, the other nodes are the schemas of the relations joined with r^C , and the labels contain the join conditions.

To keep the presentation simpler, we assume that each relation appears in the join tree at most once. Note that the specification of T is often useless, because its structure is implied by $cond$ and by the foreign keys. In the rest of the paper T is always omitted.

The quality of (contextual) association rules is evaluated by means of *support* and *confidence*. The support of the rule $(C \rightarrow cond, T)$ defined on the contextual relation r^C is the number of tuples of r^C satisfying both C and $cond$, while its confidence is the fraction of tuples associated with context C that satisfy $cond$. The association rule mining process is usually divided into two subtasks: 1) find all the sets of items (*itemsets*) whose support exceeds a given threshold *minsup*, where in our domain an item is an attribute-value pair; 2) generate, starting from the mined itemsets, the rules with a confidence greater than a specified threshold *minconf*. The first task is the most complex one, and the state-of-the-art algorithm to realize it is FP-growth [9].

As an example, a possible rule in the log of purchased movies LOG_MOVIE is $r_1 \equiv \text{user} = \text{adult} \wedge \text{time} = \text{daytime} \rightarrow \text{director.citizenship} = \text{'Italian'} \wedge \text{actor.citizenship} = \text{'French'}$; the join tree of the rule is represented in Fig. 2. Note that this rule requires joins in the opposite direction with respect to the foreign keys, and that several actors may correspond to the same movie; the condition is satisfied by all the movies in the log with Italian director and starring *at least one* French actor.

2.1 Issues in Contextual Rule Mining

In this section we describe the issues related to contextual association rule mining in relational databases. First, we explain that traditional methods cannot cope with rules with joins following the foreign keys in the opposite direction. Then, we illustrate the three main causes that lead to the extraction of several not significant rules: rule well-formedness, functional dependencies, and other kinds of key-related dependencies. A naive solution to tackle the latter problems consists in eliminating the not significant rules with a post-processing phase, but the number of generated patterns can easily

become unmanageable, therefore it is needed to avoid not only the extraction of the not well-formed contextual rules, but also of the itemsets that cannot lead to any valid rule.

Extraction of Rules with Joins Following the Foreign Keys in the Opposite Direction. Classic techniques, like FP-growth, mine frequent itemsets on *transaction databases* formed by a series of transactions, where a transaction is a set of items. A relational table can be converted into a set of transactions, where each tuple is represented by a transaction containing an item for each attribute-value pair. Such a transaction can be extended including the attributes of the relations reachable through foreign keys, e.g. DIRECTOR. The transaction could be further extended by adding also information obtained following the foreign keys in the opposite direction – e.g., ACTOR, through MOVIE_ACTOR –, but this would lead to the extraction of rules with a different meaning with respect to that indicated by the definition of contextual association rule. Consider our running example: the transaction associated with a movie purchase could be enriched with items describing the name, citizenship and gender of all the actors playing in the movie. In this way, an itemset like $\text{time} = \text{night} \wedge \text{actor.citizenship} = \text{'Italian'} \wedge \text{actor.gender} = \text{'male'}$ could be mined, along with the rule $\text{time} = \text{night} \rightarrow \text{actor.citizenship} = \text{'Italian'} \wedge \text{actor.gender} = \text{'male'}$. However, this rule is mined when users at night have often requested movies starring at least an Italian actor and at least a male actor; on the contrary, according to the definition of contextual association rule, such a rule should indicate that users at night have often requested movies with at least an actor that is *both male and Italian*.

Well-Formed Rules. Standard techniques for association rule mining extract rules where each possible item may appear both in the antecedent and in the consequent. On the contrary, the antecedent of a contextual association rule may contain only contextual attributes, and its consequent only data attributes. Consider, for example, the rule $r_2 \equiv \text{situation} = \text{alone} \wedge \text{log_movie.genre} = \text{'comedy'} \rightarrow \text{director.name} = \text{'WoodyAllen'}$: it can be mined from a contextual relation, but is not well formed.

Functional Dependencies. Relational schemas may contain functional dependencies, that cause the extraction of many *equivalent* rules, i.e. rules that always hold in the same circumstances, thus having the same support and confidence and carrying the same information. Given a functional dependency $X \rightarrow Y$, being X and Y sets of attributes, all the rules involving in their consequent all the attributes in X and at least one attribute of Y are equivalent. Keys and foreign keys generate special kinds of functional dependencies. For instance, the primary key `movie_id` of MOVIE generates a functional dependency associated with the LOG_MOVIE contextual relation, involving attributes of MOVIE as well as attributes of DIRECTOR: $\text{log_movie.movie_id} \rightarrow \text{log_movie.name} \wedge \text{log_movie.genre} \wedge \text{log_movie.director} \wedge \text{director.director_id} \wedge \text{director.name} \wedge \text{director.citizenship}$. Such a dependency makes equivalent a large number of rules; e.g.: $\text{user} = \text{adult} \rightarrow \text{log_movie.movie_id} = \text{'m1'}$, $\text{user} = \text{adult} \rightarrow \text{log_movie.movie_id} = \text{'m1'} \wedge \text{log_movie.genre} = \text{'comedy'}$.

Other Kinds of Key-Related Dependencies. Keys generate redundancies not only when they are connected to functional dependencies, but more in general every time a condition is expressed in the subtree of the join tree rooted in the relation whose key is involved in the rule, whether there is a functional dependency or not. Consider the rule $r_3 \equiv \text{situation} = \text{alone} \rightarrow \text{log_movie.movie_id} = \text{'m1'} \wedge \text{actor.citizenship}$

= 'Italian'; such a rule can be associated with the join tree in Fig. 2, excluding the DIRECTOR relation. r_3 is equivalent to $r_4 \equiv \text{situation} = \text{alone} \rightarrow \text{log_movie.movie_id} = \text{'m1'}$, even if there is not a functional dependency between the movie identifier and the actor citizenship. Equivalences of this kind arise when rules contain joins realized following the foreign keys in the opposite direction.

3 Mining Contextual Association Rules in Relational Databases

Now we review two existing techniques for mining (non-redundant) association rules in relational databases, applying them to our problem.

3.1 Itemset Mining Using FP-Growth with Constraints

The extraction of itemsets imposing constraints on the features of the discovered patterns has been thoroughly analyzed in the literature [1,10,13]. Typical constraints that have been added to FP-growth concern aggregate values computed on the items in an itemset. Consider items with a price: a possible constraint could prescribe that the sum of the prices of the items in the itemset is greater than 100, or that the maximum price of the products does not exceed 200. In the literature two categories of constraints with favorable properties have been identified: antimonotone and succinct constraints.

A constraint is **antimonotone** if, when it is violated by an itemset, it is violated also by all its supersets. The constraint enforcing the minimum support is antimonotone, as well as the one imposing that the sum of the item prices must be greater than 100.

In few words, a constraint is **succinct** if there exists a set \mathcal{I} of itemsets such that the set of itemsets satisfying the constraint can be expressed through unions and intersections of the powersets of the itemsets in \mathcal{I} . For instance, consider the constraint $\max(\text{itemprice}) > 100$, and let Item be the set of all the items and $\text{Item}_{\leq 100}$ the set of the items whose price is less than or equal to 100. The set of the itemsets satisfying the constraint is $2^{\text{Item}} \setminus 2^{\text{Item}_{\leq 100}}$, i.e., all the itemsets except those constituted exclusively by items whose price is less than or equal to 100.

The FP-growth algorithm can be summarized in the following two steps: 1) the transaction database is scanned finding the set \mathcal{FI} of individual items whose support is greater than minsup ; 2) for each $x_i \in \mathcal{FI}$, a projected database is generated including only the transactions containing x_i . Then, for each projected database, the algorithm is applied recursively. Note that when the algorithm finds at step 1 frequent items on the projected database associated with x_i , i.e. with *prefix* x_i , it is actually mining frequent itemsets of length 2, involving x_i and the newly discovered frequent item.

[13] has modified the algorithm to manage antimonotone constraints: after a frequent item b is found in a projected database with prefix a , before generating the projected database with prefix ab , the itemset ab is tested for constraint satisfaction. If the constraint is not satisfied, the new projected database is not created. Succinct constraints are considered in [10]. They assume that the set of items in presence of these constraints can be partitioned into a set of mandatory items and a set of optional items, and that the constraint is satisfied by all the itemsets containing at least one mandatory item. FP-growth is modified in such a way that, when its step 2 is executed for the very first

time, only projected databases for the mandatory items are considered; in this way, it is guaranteed that all the itemsets that are produced contain at least one mandatory item.

Using Constrained Itemsets to Discover Contextual Association Rules. In order to mine contextual association rules in relational databases, two constraints have to be imposed in the itemset mining phase. First of all, itemsets must allow the generation of rules with context in the antecedent and data in the consequent; second, itemsets containing functional dependencies must be discarded.

About the first constraint, only the itemsets involving exclusively data items must be discarded, because those containing only contextual items are useful to compute the rule confidences. This constraint is not antimonotone: the itemset $\text{log_movie.genre} = \text{'comedy'} \wedge \text{director.director_id} = \text{'250'}$ violates the constraint, but its superset $\text{log_movie.genre} = \text{'comedy'} \wedge \text{director.director_id} = \text{'250'} \wedge \text{time} = \text{night}$ does not. However, the constraint is succinct: being $Item$ the set of all the items, and $Item_d$ its subset including only the items describing conditions on data, the set of the valid itemsets with respect to the constraint can be expressed as $2^{Item} \setminus 2^{Item_d}$.

The constraint related to functional dependencies must assure that, given a functional dependency $X \rightarrow Y$, no itemsets including conditions on all the attributes of X and on at least one attribute of Y are mined. This constraint is clearly antimonotone: an itemset that is not valid (e.g., $\text{time} = \text{night} \wedge \text{director.director_id} = \text{'250'} \wedge \text{director.citizenship} = \text{'American'}$) cannot become valid adding further items. On the contrary, it is not possible to identify sets of compulsory and optional items, therefore the optimizations of [10] for succinct constraints cannot be applied.

To summarize, incorporating our two types of constraints into FP-growth requires the following modifications: 1) The very first time that step 2 is executed, only the projected databases of the context items are generated. 2) Before generating a projected database with prefix I , the itemset I is checked for redundancy against the functional dependencies; if the constraint is not satisfied, such a projected database is not built.

Once the itemsets have been discovered, the contextual rule generation is straightforward: for each itemset that contains at least a condition on data, the corresponding rule is outputted keeping the context in the antecedent and the data in the consequent.

With respect to the issues described in Sect. 2.1, the technique just presented, relying on transaction databases, cannot discover rules with joins following the foreign keys in the opposite direction. On the contrary, it is possible to deal with the redundancies connected to rule well-formedness and to functional dependencies. Finally, since the rules with joins following foreign keys in the opposite direction cannot be mined, the issue related to the other key-related dependencies is not applicable here.

3.2 Mining Frequent Conjunctive Queries

Goethals et al. [8,7] propose the algorithm Conqueror⁺ to mine frequent conjunctive queries in relational databases; the algorithm does not require transaction data, but is able to operate directly on a relational database, using SQL and JDBC. This algorithm allows to discover arbitrary joins between relations, including those following the foreign keys in the opposite direction, that are interesting for our aims. Conqueror⁺ is also able to avoid the extraction of duplicate queries due to functional dependencies, computing the closure of the selection condition with respect to the functional

dependencies. An example of pattern that can emerge in our reference scenario is $\sigma_{\text{genre}='comedy' \wedge \text{actor.citiz}='Ita'} \text{LOG_MOVIE} \bowtie \text{MOVIE_ACTOR} \bowtie \text{ACTOR}$.

The Conqueror⁺ algorithm considers all the possible joins. The operations performed for each join can be schematized as follows:

- Build a queue containing all the queries for which selection conditions have to be searched. At the beginning, it contains only the join query with no selections.
- While the queue is not empty:
 1. Extract a query Q from the queue
 2. For each attribute A not already in the selection of Q :
 - (a) Add A to the selection of Q
 - (b) If necessary, compute the closure to deal with functional dependencies
 - (c) If frequent (exceeding minsup) values for the selections of Q exist, output the queries with the frequent values assigned, and re-insert Q in the queue

Suppose we are searching for frequent queries on the relation DIRECTOR with no joins. We begin with the query without selections, and start adding conditions on attributes. Imagine that we consider first `director_id=?` DIRECTOR is generated at step 2(a). Then, at step 2(b) the closure is computed, leading to the following: $\sigma_{\text{director_id}=? \wedge \text{name}=? \wedge \text{citiz}=?} \text{DIRECTOR}$. At step 2(c) frequent values for the selections are searched, and several queries could be outputted; an example of these queries could be $\sigma_{\text{dir_id}='250' \wedge \text{name}='W.Allen' \wedge \text{citiz}='Amer'} \text{DIRECTOR}$. Because we have found frequent values for the query, we insert it again into the queue in order to search for more attributes to add; in this case no attributes are remaining, so the algorithm ends.

After that the frequent queries – that play the role of the frequent itemsets in the traditional methods – have been found, the authors suggest to generate rules $Q_1 \rightarrow Q_2$ where Q_2 is a query that is more restrictive than Q_1 .

Using Frequent Conjunctive Queries to Discover Contextual Association Rules.

We are interested only in mining rules involving contextual relations. In more detail, for each pair of frequent queries (Q_1, Q_2), the rule $Q_1 \rightarrow Q_2$ must be outputted only if: 1) Q_1 contains only conditions on context attributes; 2) Q_2 is more restrictive than Q_1 and contains also conditions on data attributes. An example of rule in our scenario, indicating that the users often watch thrillers at night is $Q_1 \rightarrow Q_2$, with $Q_1 = \sigma_{\text{time}='night'} \text{LOG_MOVIE}$, and $Q_2 = \sigma_{\text{time}='night' \wedge \text{genre}='thriller'} \text{LOG_MOVIE}$.

The above procedure can discover rules with joins following the foreign keys in the opposite direction, and can deal with functional dependencies. However, the redundancies connected to rule well-formedness are removed only during the rule generation. Finally, this algorithm discovers rules involving arbitrary joins on foreign keys, but it does not take into account the additional redundancies related to key dependencies.

4 Adding Constraints to Frequent Query Mining

The discussion carried out so far has highlighted how the solution of Goethals et al. [8,7] is the unique able to fully comply with our definition of contextual association rule, because it is the only one managing rules with joins realized following the foreign keys in the opposite direction. However, such a technique has some limitations with

respect to our aims: it is unable to impose the presence of contextual items when extracting frequent queries, and does not remove redundant rules due to keys except when functional dependencies are present. Constrained itemset mining, on the contrary, has the potentiality to deal with such issues. Now we present our proposal for mining contextual rules in relational databases enriching the extraction of frequent queries with explicit constraints: we define three constraints, dealing with the issues described in Sect. 2.1.

Constraint \mathbb{C}_1 deals with rule well-formedness imposing at least one contextual attribute in the selection of each query, and can be enforced at step 2c of Conqueror⁺: the first attribute that is added to the selection must be chosen exclusively among contextual attributes; then, the others may be indifferently contextual or data attributes.

Constraint \mathbb{C}_2 considers the redundancies related to the keys. Let $sel(Q)$ be the conjunctive selection condition of a query Q . The query must satisfy the following:

$\mathbb{C}_2(Q)$ true iff $sel(Q)$ does not contain a key of a relation $R(X)$ and a condition in the subtree of the join tree of Q rooted in $R(X)$

Such a constraint must be checked at step 2a of Conqueror⁺, before adding each new attribute to the selection condition.

Note that constraint \mathbb{C}_2 also deals with some redundancies caused by functional dependencies, and in particular those in which the functional dependency is due to a key. It is possible to employ constraint \mathbb{C}_3 to manage the remaining functional dependencies, thus avoiding step 2b of Conqueror⁺. \mathbb{C}_3 is defined as follows:

$\mathbb{C}_3(Q)$ true iff $sel(Q)$ does not contain all the attributes in the left side of a functional dependency and at least one attribute in the right side

Algorithm 1 summarizes our proposal for mining frequent queries for the generation of well-formed and non-redundant contextual association rules on a contextual relation. Consider for example the join containing only the LOG_MOVIE relation. At Line 2 the query without selections is enqueued, and at Line 6 the set *Att* is filled with the contextual attributes. Suppose that the first attribute in *Att* is *time*, leading to the query $\sigma_{time=?}LOG_MOVIE$. The constraints at Line 12 are obviously satisfied, since no other selection conditions have been included yet, so frequent values for *time* are searched and the connected queries outputted (Line 14). For example, an outputted query might be $\sigma_{time='night'}LOG_MOVIE$. Then, the query $\sigma_{time=?}LOG_MOVIE$ is enqueued (Line 15), ready to be extended with further attributes in the selection condition.

After that the frequent queries have been discovered, the rule generation step may be realized as described in Sect. 3.2.

5 Experimental Results

As shown in the paper, constrained itemsets can deal only with a subset of the patterns we are interested in, therefore the experiments focus on the comparison between our proposal and the basic Conqueror⁺. We have implemented our solution by extending the publicly available Java implementation¹ of Conqueror⁺. As in [8,7], the evaluation has been concentrated on the most complex phase, i.e. the frequent query mining.

¹ Downloadable at <http://adrem.ua.ac.be/conqueror>

Algorithm 1. Frequent Query Mining for Contextual Association Rules

Input: Contextual relation $R(X)$, list of functional dependencies
Output: Set of useful, non-redundant frequent queries

```

1: for all possible joins do
2:   Queue = {Join query without selections}
3:   while Queue is not empty do
4:     Q = pop(Queue)
5:     if sel(Q) is empty then
6:       Att = contextual attributes not yet evaluated for this query
7:     else
8:       Att = all the attributes of the joined relations not yet evaluated for this query
9:     for all A ∈ Att do
10:      Add A to sel(Q)
11:      Build the join tree of Q
12:      if C2(Q) ∧ C3(Q) then
13:        if frequent values for sel(Q) exist then
14:          Output the frequent queries
15:          Add Q to the queue

```

The experiments have been performed on our video-on-demand dataset extended with two synthetic relations in order to properly evaluate the capability of our proposal of dealing with redundancies related to joins against the foreign keys. In more detail, a synthetic bridge relation connects each director with exactly 15 entries of a second synthetic relation. This synthetic relation might represent, for instance, events the directors participated in. The schema of the database is reported below, along with the cardinalities of the tables. The contextual relation is LOG_MOVIE, which has been populated by a real user who queried the considered database.

```

LOG_MOVIE(id, day, time, situation, movie_id, title, genre) (4773 rows)
DIRECTOR(director_id, name) (1607 rows)
ACTOR(actor_id, name) (2897 rows)
MOVIE_DIRECTOR(movie_id, director_id) (3102 rows)
MOVIE_ACTOR(movie_id, actor_id) (6023 rows)
SYNTH(synth_id, synth_attr) (100 rows)
DIRECTOR_SYNTH(synth_id, director_id) (24105 rows)

```

We have mined frequent queries with Conqueror⁺ and with our extension for different minimum support values, and we have measured the number of mined patterns and the execution time. The experiments have been realized with an Intel Core 2 Duo CPU with 2.5 GHz and 3 GB main memory, running the PostgreSQL 8.3 DBMS. The results are shown in Table 1. As it was largely expected considering the discussion developed in the paper, our extended algorithm, when dealing with contextual rule mining, extracts less queries with respect to Conqueror⁺, because it only discovers the necessary ones. This also leads to better execution times, even of almost an order of magnitude when the minimum support is low. Clearly, the extent of the improvement heavily depends on the

Table 1. Experimental results related to frequent query mining

		Minsup	20	50	100	150	200	500
Conqueror ⁺	#queries	281510	97437	63315	53959	53637	10304	
	Execution time (sec)	46585	28909	8179	4993	4265	853	
Our extension	#queries	49278	26725	20323	18809	18621	3649	
	Execution time (sec)	6058	1879	1662	1530	1494	684	

prone to generate queries that are filtered by our constraints; for instance, adding a joined synthetic relation to ACTOR, as we have done with the directors, would obviously further widen the gap between the two compared algorithms.

6 Conclusion

In this paper the problem of contextual association rule mining in relational databases has been defined, highlighting the related challenges. Two association rule mining techniques of the literature have been compared and a novel algorithm merging the two solutions has been proposed; preliminary results have been shown.

Since our main research goal is context- and preference-based data tailoring, we are currently applying the novel algorithm introduced in this paper in our wider framework [12,11] for context-aware preference mining and relational database personalization.

References

1. Appice, A., Berardi, M., Ceci, M., Malerba, D.: Mining and filtering multi-level spatial association rules with ARES. In: Hacid, M.-S., Murray, N.V., Raš, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 342–353. Springer, Heidelberg (2005)
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. *Int. Journal of Ad Hoc and Ubiquitous Computing* 2(4), 263–277 (2007)
3. Baralis, E., Cagliero, L., Cerquitelli, T., Garza, P., Marchetti, M.: Context-aware user and service profiling by means of generalized association rules. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) KES 2009, Part II. LNCS, vol. 5712, pp. 50–57. Springer, Heidelberg (2009)
4. Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F.A., Tanca, L.: A data-oriented survey of context models. *SIGMOD Record* 36(4), 19–26 (2007)
5. Bolchini, C., Quintarelli, E., Tanca, L.: CARVE: Context-aware automatic view definition over relational databases. *Inf. Syst.* 38(1), 45–67 (2013)
6. Cremonesi, P., Garza, P., Quintarelli, E., Turrin, R.: Top-N recommendations on unpopular items with contextual knowledge. In: *Proc. of CARS. CEUR-WS.org* (2011)
7. Goethals, B., Laurent, D., Le Page, W.: Discovery and application of functional dependencies in conjunctive query mining. In: Bach Pedersen, T., Mohania, M.K., Tjoa, A.M. (eds.) DAWAK 2010. LNCS, vol. 6263, pp. 142–156. Springer, Heidelberg (2010)
8. Goethals, B., Le Page, W., Mannila, H.: Mining association rules of simple conjunctive queries. In: *Proc. of SDM*, pp. 96–107. SIAM, Philadelphia (2008)
9. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.* 8(1), 53–87 (2004)
10. Leung, C.K.-S., Lakshmanan, L.V.S., Ng, R.T.: Exploiting succinct constraints using FP-trees. *SIGKDD Explorations* 4(1), 40–49 (2002)
11. Miele, A., Quintarelli, E., Rabosio, E., Tanca, L.: A data-mining approach to preference-based data ranking founded on contextual information. *Inf. Syst.* 38(4), 524–544 (2013)
12. Miele, A., Quintarelli, E., Tanca, L.: A methodology for preference-based personalization of contextual data. In: *Proc. of EDBT*, pp. 287–298. ACM Press, New York (2009)
13. Pei, J., Han, J., Lakshmanan, L.V.S.: Pushing convertible constraints in frequent itemset mining. *Data Min. Knowl. Discov.* 8(3), 227–252 (2004)
14. Stefanidis, K., Pitoura, E., Vassiliadis, P.: Managing contextual preferences. *Inf. Syst.* 36(8), 1158–1180 (2011)

Challenges and Issues on Collecting and Analyzing Large Volumes of Network Data Measurements

Enrico Masala², Antonio Servetti²,
Simone Basso^{1,2}, and Juan Carlos De Martin^{1,2}

¹ NEXA Center for Internet & Society

² Internet Media Group

Control and Computer Engineering Department

Politecnico di Torino, 10129 Torino, Italy

`firstname.lastname@polito.it`

Abstract. This paper presents the main challenges and issues faced when collecting and analyzing a large volume of network data measurements. We refer in particular to data collected by means of Neubot, an open source project that uses active probes on the client side to measure the evolution of key network parameters over time to better understand the performance of end-users' Internet connections. The measured data are already freely accessible and stored on Measurement Lab (M-Lab), an organization that provides dedicated resources to perform network measurements and diagnostics in the Internet. Given the ever increasing amount of data collected by the Neubot project as well as other similar projects hosted by M-Lab, it is necessary to improve the platform to efficiently handle the huge amount of data that is expected to come in the very near future, so that it can be used by researchers and end-users themselves to gain a better understanding of network behavior.

Keywords: Network Data Collection, Network Data Analysis.

1 Introduction

In the study of the performance of the Internet one of the major challenges is how to collect a large number of reliable network measurements from a sufficiently large number of locations in the network [1]. In the recent years, several independent tools have been developed to collect this type of information [2–4].

To study and monitor the performance of Internet access links, we designed Neubot [5–7], a software for distributed network measurements that runs on the user's computer and periodically monitors the performance of its connection to the Internet. The results are collected on the user device and made publicly available on a central server to allow constant monitoring of the state of the Internet by any interested parties.

Since Feb 9, 2012, Neubot has been hosted on Measurement Lab (M-Lab) [8, 9], an organization that provides dedicated server-side resources to open-source

network measurement tools, including Neubot and NDT [4]. The availability of a network of servers around the world allows Neubot to effectively test the performance of the clients broadband access network, by connecting them to the closest M-Lab server available. In addition, in the near future, we will release a mobile version of Neubot that will be available for the increasing number of mobile Internet devices, which is predicted to surpass the number of desktop devices by the end of 2014 [10].

Thus, while Neubot is performing a large amount of measurements each day, the problem of measuring the network is becoming a problem of managing the available data for storage, querying and analysis purposes. For the first two challenges Neubot may take advantage of two services provided by M-Lab, namely the Google Cloud Storage [11] and the Google Big Query platform [12]. However, a more flexible solution is needed to perform a deeper analysis of the measurements and to gain clear, if possible, and real-time insight into the behavior of the Internet and of the Internet connection. of the end users.

The objective is to be able to answer questions such as the following: Is the server on the other end of the connection having a problem? Is my device or modem not properly configured? Is something wrong with the ADSL connection? Is an ISP deliberately interfering with my traffic? Only a flexible platform that can efficiently manage the processing of measures collected by millions of users that share the same (or similar) network path could answer these questions. In fact, benchmarking Internet access link performance cannot be achieved by merely running a single speed test. Speed varies with time and it is affected by a number of confounding factors (i.e., home network cross-traffic, end-host configuration, wireless connection quality) that must be isolated as much as possible by proper data processing and analysis.

The rest of this paper is organized as follows. In Section 2 we present the Neubot architecture for data collection and we describe the tests currently implemented. Information about data query and data analysis is given respectively in Section 3 and in Section 4. Finally conclusions are drawn in Section 5 and possible future work is presented.

2 Architecture for Data Collection

The architecture of Neubot consists of an *agent* that runs on the users' computer, and of a set of servers as shown in Fig. 1. The servers have different roles and may be replicated in different locations of the M-Lab network. We distinguish between the *master server*, the *test servers* and the *archive server*.

The Neubot *agent* runs in background as a system service and periodically check the master server to be informed on the next test to perform and with which test server.

The *master* server acts as a coordinator. Once contacted, it can implement different policies for the coordination of the Neubot agents. For example, for certain kind of tests that aim at measuring the speed of the user's connection to the Internet (e.g., *speedtest*), it can select the test server that is closer to the

agent, for another kind of tests, that do not have particular requirements (e.g., *raw*), it can select the test server with the lowest load.

The *test* servers implement one or more transmission tests. First, a negotiation phase assigns a temporary unique identifier to each connecting agent that wants to perform a test, and uses this identifier to manage the queue of incoming test requests, i.e., to schedule each single test when the right conditions are met. Second, a test phase implements the actual transmission test to estimate selected characteristics of the network between the agent and the test server. The measured performance metrics depend on the type of test and are described later. Once the test is completed, the agent uploads the results to the test server, and saves a copy locally.

The *archive* server once a day collects the test results from each test server and makes them available on the web.

M-Lab Support. The test servers are hosted on the M-Lab network [9] in 114 different servers operating across 32 geographically distributed sites around the globe (as of April 2013).

Each Neubot server instance is allocated dedicated resources on the M-Lab platform to facilitate accurate measurements and perform high-bandwidth measurements at large scale (e.g., each server is connected to one or more Internet Service Providers with 1 Gbit/s upstream capacity).

The archive server is hosted on the Google Cloud Storage service (similar to Amazon S3) [11]. Currently, Neubot data is uploaded to this public repository in batches once a day. Raw data are organized into tarballs that contain all the data collected during a single day on a single M-Lab server.

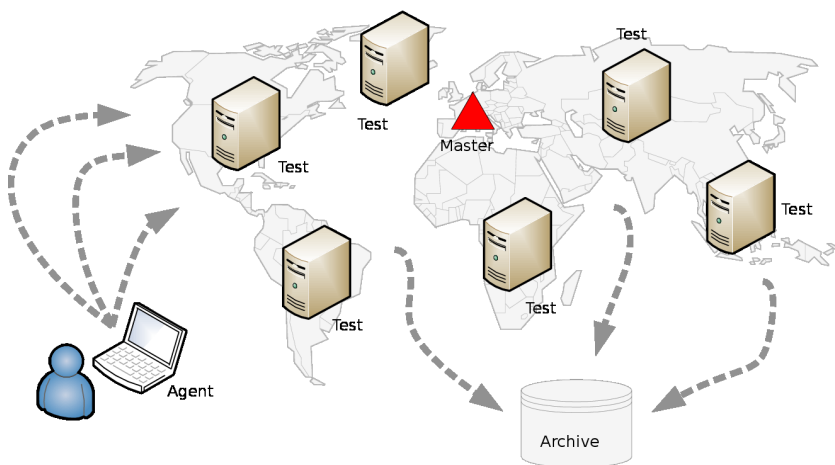


Fig. 1. Neubot architecture. Agents on the users computer are coordinated by the Neubot master server and connect to the preferred test server to perform the actual tests. Results are collected on the archive platform.

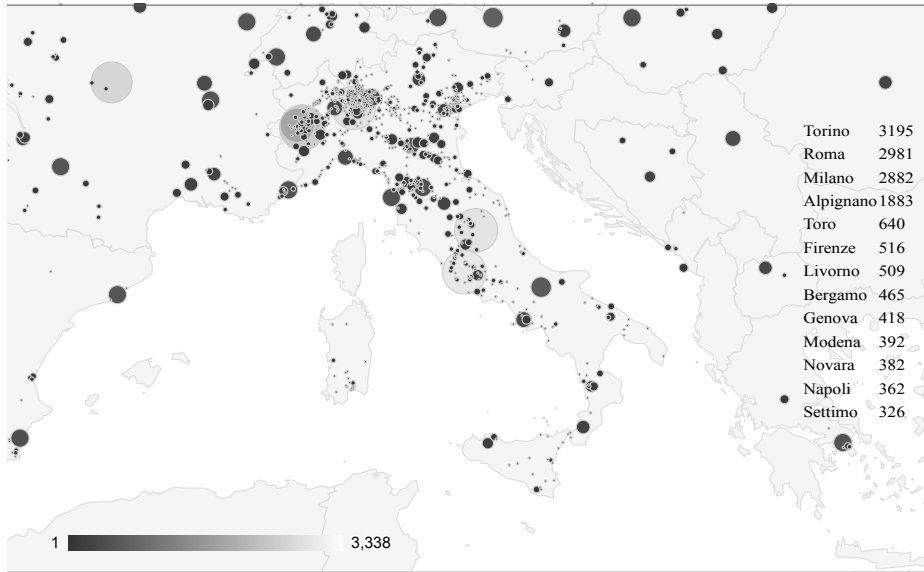


Fig. 2. Geographical distribution of the Neubot clients in Italy on the basis of their IP address. The number of test performed from each site is presented in the table on the right and corresponds to the color and size of the circles in the figure. Figure refers to tests run in April 2013.

In a month about 500'000 tests are run and the collected data sums to about half a gigabyte (gzip compressed). Fig. 2 shows the location of the clients and the number of tests run on April 2013 in Italy. A more extended representation covering all the 10'000+ different IPs and the 2'000+ different locations from which Neubot test have been performed (e.g., Singapore, Rome, Cape Town, Toronto, etc.) can be accessed on the web at the following link http://bit.ly/neubot_gmap. The amount of data collected by Neubot could considerably increase if the project becomes more widely known as, for instance, is the case of the Network Diagnostic Tool (NDT) [4], another tool available on M-Lab, that collects up to 1 TB of compressed data each month.

2.1 Implemented Tests

Neubot implements three active network tests: *bittorrent*, *raw* and *speedtest*. Neubot schedules one of these tests at random every half an hour. In addition the user has the possibility of running tests on demand.

The speedtest test emulates HTTP and estimates the round-trip time, the download and the upload bandwidth. The bittorrent test emulates BitTorrent peer-wire protocol and estimates the round-trip time, the download and the upload bandwidth. The raw test performs a raw 10-second TCP download to estimate the download bandwidth, and it collects statistics about the TCP

sender by using Web100 [13]. Test results are saved in JSON format and range from 470 to 30,000 bytes per test (the latter being the raw test that collects data periodically every 10 seconds).

Among the data collected for each speedtest test there are:

- uuid** Random unique identifier of the Neubot instance, useful to perform time series analysis.
- timestamp** Time when the test was performed, expressed as number of seconds elapsed since midnight of January 1, 1970.
- real_address** Neubot's IP address, as seen by the server. It is either an IPv4 or an IPv6 address.
- remote_address** The server's IP address. It is either an IPv4 or an IPv6 address.
- connect_time** Round-trip time (RTT) estimated by measuring the time that the connect system call takes to complete, measured in seconds.
- download_speed** Download speed measured by dividing the number of received bytes by the download time, measured in bytes per second.
- upload_speed** Upload speed measured by dividing the number of sent bytes by the upload time, measured in bytes per second.
- latency** RTT estimated by measuring the average time elapsed between sending a small request and receiving a small response, measured in seconds.

2.2 Expected Evolution

A significant effort is currently being devoted to port the Neubot client to a mobile platform, i.e., Android. This will greatly extend the amount of collected data since the availability as an app will increase the chance that the application is installed on mobile devices. Similar applications already exists, such as speedtest.net, which had between 10 and 50 millions of downloads from the Android market (i.e. between 1/100 and 1/20 of the number of Android users [14]). If 1 out of 1,000 Android users installs the Neubot application, between 1 and 10 million tests could be performed each day, considering that Neubot run tests periodically. Moreover, new types of tests such as HTTP-based ones will be soon implemented, which will further increase the amount of collected data. To analyze such large amount of data, which will be 100 to 1,000 times the current amount, new analysis techniques will be necessary. The potential issues arising from this scenario are described in the remaining part of this paper.

3 Data Querying

The first objective of a data collection effort such as the described one is to be able to look into the data themselves to gain some understanding about both the data collection process and the statistics extracted from the data. Although currently the amount of data is somewhat limited, i.e., in the order of few

gigabytes, it is expected that this amount will increase rapidly as the software used for data collection will be ported to mobile platforms. In such conditions it is very important to be able to monitor some key parameters of the system, such as the number of unique users, how they move, how frequently they change IP (or provider), etc.

The above monitoring activity, however, cannot be performed directly on the test servers. To avoid to perturb and/or invalidate the network experiments, in fact, M-Lab servers are dedicated to the measurement task only. Therefore, the collected data must be moved to a different location where they are permanently stored, and where they can be queried and analysed.

In particular, the current implementation of M-Lab moves test results from each test server to a public repository, i.e., the *archive* server, once a day. The repository is hosted by the Google Cloud Storage service. In this way the compressed archives of the test results can be accessed through a web interface or by means of the Google Cloud Storage command line utility.

The designers of M-Lab decided to use compressed archives to efficiently collect and transfer the data in a public location, but the whole range of data are only useful to those planning to run an extensive and detailed analysis. For most researchers aggregated information may be sufficient. Moreover most people cannot afford either the time required to download or the amount of TB necessary to store such large amount of data for personal processing (several TBs when the data coming from all the M-Lab tools are included). In addition, a large amount of processing power is required to manage this volume of data.

Thus a different structure has been envisaged for the results collected on M-Lab (including Neubot) to be able to share them with the community of the researchers and allow querying without downloading. As it is already the case for other M-Lab tools such as NDT, in the future the data will be made available through the Google BigQuery web service that allows to run interactive analysis (e.g., SQL queries) of large datasets [12].

Interested parties will then be able to query the measurement data in a matter of seconds – even with complex queries – in order to gain a better understanding into Internet operation and performance. For example, real-time processing of millions of measurements may allow to identify Internet congestion, traffic shaping, or network outages on a world scale.

4 Data Analytics

The large amount of measurement data already available allows to employ data mining techniques to discover correlations among data that would otherwise be difficult to observe. In this section we present an overview of the type of analysis useful for the purpose of analyzing the network behavior both as a whole and with respect to the experience of each single user. Examples of possible analyses are given by showing some small but representative subsets of data.

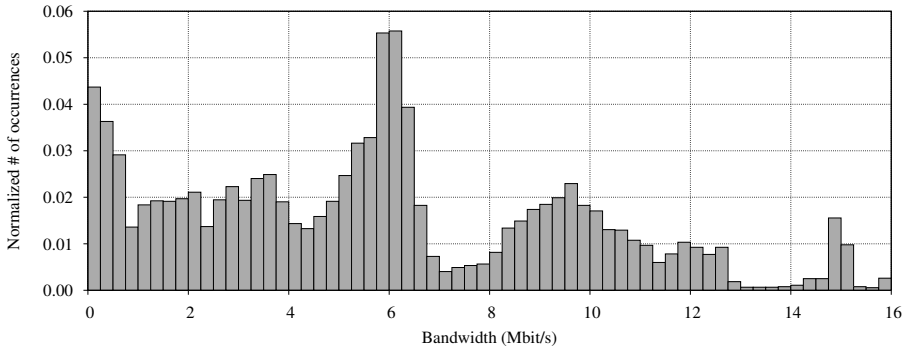


Fig. 3. Normalized number of occurrences of download speed values. The plot considers only clients connected using a given Internet operator.

4.1 Dimensions of the Analysis

First, note that the data presents several dimensions that require careful analysis. The most important ones are highlighted in the following.

time: clients run tests periodically when connected to the Internet. The evolution over time of the measured parameters need to be considered to better understand the situation of the connection and relate it to other clients in similar conditions at the same time instant.

location: the information about the approximate physical location of the client will play a role in understanding if any unusual value detected in the Internet access parameters is due to the location (e.g., scarcity of provisioned resources in a certain area) or not (e.g., limitations imposed by the provider). This can be detected by analyzing the behavior of the parameters for other clients in the same conditions.

network metrics: the values of the network metrics themselves need to be analyzed, since the active measurements are very informative but they can be influenced by the concurrent usage of other network-based applications.

connectivity type: the widespread use of wireless technology for Internet access will require to consider the parameters differently depending on the connection type, in order not to mix the data coming from wired connection with wireless ones, since the constraints and business models behind the provisioning of such connections are widely different.

Note that some types of data may not be readily available and they potentially need to be inferred by others. For instance, recognizing the connection type is not easy when only network-level metrics are available.

4.2 Challenges of the Analysis and Potential Solutions

Data analytics approaches need to be used both to recognize and cluster together similar behaviors with particular attention to the dimensions mentioned before. A possible analysis is described here to give hints about the type and complexity of the data processing needed.

An important aim of the Neubot and similar projects [15, 16] is to understand if events are local to the user or they happen as the result of more general conditions, for instance network congestion. In fact, in the first case the user would probably like to be informed to better understand if the behavior is normal and it conforms to his/her expectations, whereas in the latter it is highly probable that some activity between the user and the Internet backbone is happening (e.g., traffic shaping).

To detect this condition, a possible approach is described in the following. First, some statistical performance analysis should be done to know if the experienced performance is typical or it happens only sporadically. This type of processing may be demanding in terms of computational resources, especially if the data need to be clustered. For instance, data can be separated by operator on the basis of the client IP address, and within each subset a number of clusters need to be identified, which correspond to the typical connection speeds offered to the users. An example of such a situation is shown in Fig. 3. The data refers to clients whose IP belong to a given network operator. Data suggest that there are some download speeds more common than others: in fact they correspond to the typical commercial DSL offers of that operator. However, due to the number of operators in the world, it would be infeasible to analyze the data manually to find those clusters, hence an automatic approach is needed.

Also, note that the dimensions mentioned before can influence the position of the clusters, for instance the average value of the cluster depends on many factors, such as time and location. Moreover, intercorrelation between those average values may be present and they need to be searched since they can be a very valuable indication to researchers to understand network dynamics.

4.3 Case Study

Once the previous type of analysis is available, users whose measurements significantly differ from the expected behavior can be alerted in real-time. To show the possibilities offered by such approach we present a simple analysis conducted on a limited amount of data. Fig. 4 shows the download speed of all tests run in April 2013 from clients connected to the Internet using a given operator, as grey points. The data also show the time of the day at which the test was run. The density of grey points is correlated to the probability that certain download speed values are measured in the tests. A specific user is also plotted in the graph using black asterisks. Most of its points are close to the center of one of the clusters, i.e., around 750 kbit/s. However, some of them are quite far. This is the type of situations which are deemed interesting, both from the research and the user's point of view. Researchers may be interested in understanding why this

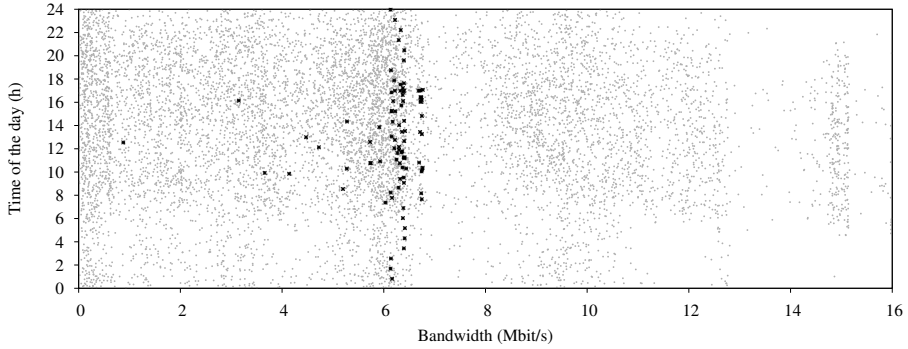


Fig. 4. Download speed of a specific user (black asterisks) over all the tests (grey dots) performed by users using the same operator for Internet connection.

happens, while affected users could simply be warned about the unusual values measured by the tool, so they can decide if it is important to further investigate the situation or not. Note that this simple example consider only one user to simplify visualization, but algorithms should be able to consider more users at a time to investigate if the behavior is typical or due to some specific, isolated, reasons. Moreover, data analysis and clustering must also be run in real-time since the unusual behavior may be due only to transient reasons that however affect many users, e.g., congestion in the network. This is important since it allows to distinguish typical behaviors and patterns from transient conditions.

In any case, a scalable approach should be used so that even complex algorithms such as machine learning or data mining ones can be efficiently run on large set of data. Moreover, results should be obtained quickly enough to be useful to the users of the system, e.g., informing them about the characteristics of the detected situation. A possible approach could be to employ libraries such as the one of the Mahout project [17] that promises to provide scalable algorithms for these purposes.

5 Conclusion and Future Work

This work presented the main challenges and issues faced when collecting and analyzing a large amount of network data measurements. The data collection architecture of the Neubot project has been discussed including potential evolutions. The possibility to collect huge amount of data has been addressed from the point of view of both querying data and analyzing it with more complex algorithms, potentially in real-time. The algorithms that are expected to be used on such data have been discussed including their scalability implications and how to efficiently address them. Data mining algorithms such as association rules may be successfully applied to discover interesting correlations that are hidden

in the data and that will help in the knowledge discovery process. For example, the analysis of data distribution over time and for different providers may be exploited to identify periods of time or providers that become slower or less reliable more frequently than usual. We believe that the ability to process such huge amount of data with complex algorithms in real time can greatly contribute to gain more understanding in network dynamics by researchers interested in the area as well as by end-users interested in knowing additional information about the conditions of the network to which they are connected.

References

1. Palfrey, J., Zittrain, J.: Better data for a better Internet. *Science* 334(6060), 1210–1211 (2011)
2. Sundaresan, S., de Donato, W., Feamster, N., Teixeira, R., Crawford, S., Pescapè, A.: Broadband internet performance: a view from the gateway. *ACM SIGCOMM Computer Communication Review* 41(4), 134–145 (2011)
3. Speedtest.net: The global broadband speed test, <http://speedtest.net>
4. Carlson, R.: Network diagnostic tool, <http://e2epi.internet2.edu/ndt/>
5. Neubot: The network neutrality bot, <http://neubot.org/>
6. Basso, S., Servetti, A., De Martin, J.C.: Rationale, design, and implementation of the network neutrality bot. In: Proc. of AICA, L'Aquila, Italy (September 2010)
7. Basso, S., Servetti, A., De Martin, J.C.: The network neutrality bot architecture: a preliminary approach for self-monitoring of Internet access QoS. In: Proc. of IEEE 16th International Symposium on Computers and Communications (ISCC), Corfu, Greece (July 2011)
8. Measurement Lab, <http://www.measurementlab.net/>
9. Dovrolis, C., Gummadi, K., Kuzmanovic, A., Meinrath, S.D.: Measurement lab: Overview and an invitation to the research community. *ACM SIGCOMM Computer Communication Review* 40(3), 53–56 (2010)
10. Meeker, M.: Internet trends. In: D10 Conference, Rancho Palos Verdes, CA, USA (May 2012)
11. Google: Cloud Storage, <https://developers.google.com/storage/>
12. Google: Big Query, <https://developers.google.com/bigquery/>
13. Web 100 project, <http://www.web100.org/>
14. Asymco: When will Android reach one billion users? (February 2012), <http://www.asymco.com/2012/02/29/when-will-android-reach-one-billion-users/>
15. Spring, N., Wetherall, D., Anderson, T.: Reverse engineering the Internet. *ACM SIGCOMM Computer Communication Review* 34(1), 8 (2004)
16. Kanuparth, P., Dovrolis, C.: ShaperProbe: end-to-end detection of ISP traffic shaping using active methods. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, pp. 473–482. ACM (2011)
17. Apache: Mahout project (2013), <http://mahout.apache.org>

Part III

**The Second International
Workshop on
GPUs in Databases**

GPU-Accelerated Query Selectivity Estimation Based on Data Clustering and Monte Carlo Integration Method Developed in CUDA Environment

Dariusz Rafal Augustyn and Lukasz Warchal

Silesian University of Technology, Institute of Informatics,
16 Akademicka St., 44-100 Gliwice, Poland
{draugustyn, lukasz.warchal}@polsl.pl

Abstract. Query selectivity is a parameter that allows to estimate the size of data satisfying a query condition. For complex range query condition it may be defined as multi integral over a multivariate probability density function (PDF). It describes a multidimensional attribute value distribution and may be estimated using the known approach based on a superposition of Gaussian clusters. But there is the problem of an efficient integration of the multivariate PDF. This may be solved by applying Monte Carlo (MC) method which exposes its advantages for high dimensions. To satisfy the time constraint of selectivity calculation, the parallelized MC integration method was proposed in the paper. The implementation of the method is based on CUDA technology. The paper also describes the application designated for obtaining the time-optimal parameter values of the method.

Keywords: Selectivity Estimation, Data Clustering, Monte Carlo Integration, CUDA.

1 Introduction

Efficient database query executing requires from a cost query optimizer (CQO) to obtain the best execution plan during a query prepare phase. Performing the prepare phase is time-critical and commonly it is assumed that it should take no more than a few milliseconds. In this phase CQO needs to early estimate the query result size to choose the optimal way of query processing. Thus a selectivity parameter is introduced. It is the number of table rows satisfying given condition divided by the number of all rows in this table. For a single-table range query with a condition based on many attributes with continuous domain the selectivity is defined as follows:

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

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

To obtain the D -th order definite integral value a representation of multidimensional PDF is required. Because of the curse of dimensionality problem, a representation based on a simple multidimensional histogram may be not space-efficient. There are many approaches to the problem of small space-consuming multidimensional distribution representation e.g. kernel estimator [11], cosine series [12], discrete wavelets transform [3], Bayesian network [4], sample clustering [2,8], etc. Here, we use the approach which is based on a superposition of Gaussian clusters [2].

The mentioned time limit for the prepare phase forces the relevant upper limit for a selectivity estimation, which is commonly assumed as about $1ms$. This time constraint is difficult to satisfy but GPU parallel processing capabilities and CUDA technology may help to solve the problem. Such approach was already proposed for a DCT–spectrum–based representation of attribute values distribution [1] (no integration of PDF is needed in [1]). Here, we want to present the efficient, GPU-based method of selectivity estimation which operates on an approximation of a multivariate PDF. This approach requires to calculate a multi-integral, which is done with CUDA-based Monte Carlo integration method.

The contributions of this paper are:

- adapting Monte Carlo integration method to GPU for selectivity estimation based on multidimensional estimator of PDF (applying GPU may improve enough the efficiency of this method to fulfill the assumed time limitations),
- proposing algorithms for finding the time-optimal values of a method parameters (like the number of samples processed by each GPU thread).

2 Superposition of Multivariate Normal Distributions as an Approximate Representation of Multidimensional PDF – The Theoretical Background

Let us assume that a given multidimensional distribution of attribute values is well approximated by a weighted sum of Gaussian distributions. The approximation may be done by clustering data using some variants of the well-known GK method [6] or the one proposed in [2]. During the update statistics we may find the number of Gaussian clusters and weight and parameters for each of them. This allows to obtain an estimator of a multivariate PDF (see Fig.1).

Let us assume:

G – the total number of Gaussian clusters,

\mathbf{M}_k – the center of the k -th cluster for $k = 1 \dots G$ (D elements),

\mathbf{C}_k – the covariance matrix of the k -th cluster ($D \times D$ elements)

$f_k(\mathbf{x})$ – PDF of the k -th cluster:

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} (\det(\mathbf{C}_k))^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{M}_k)^T \mathbf{C}_k^{-1}(\mathbf{x} - \mathbf{M}_k)\right) \quad (2)$$

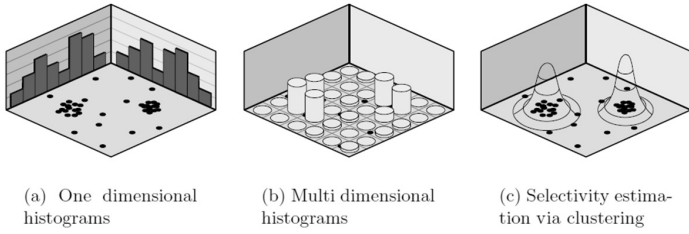


Fig. 1. Visualization of different concepts of selectivity estimation (source [2])

w_k – the weight of the k -th cluster (the number of samples which belongs to the k -th cluster divided by the total number of samples taken from database during update statistics),

$f(\mathbf{x})$ – PDF as the linear combination of $f_k(\mathbf{x})$:

$$f(\mathbf{x}) = \sum_{i=1}^G w_k f_k(\mathbf{x}). \quad (3)$$

Obtaining G , w_k , \mathbf{M}_k , \mathbf{C}_k is not done during an on-line query data processing therefore it is not time-critical. However, a selectivity calculation for a concrete query condition is. This is the reason why we consider the problem of an efficient selectivity method that operates over an already prepared PDF estimator.

Let us introduce the modified weight coefficient as follows:

$$wg_k = \frac{w_k}{(2\pi)^{\frac{D}{2}} (\det(\mathbf{C}_k))^{\frac{1}{2}}} \quad (4)$$

for $k = 1 \dots G$. Hence, the formula 3 is equivalent to:

$$f(\mathbf{x}) = \sum_{i=1}^G wg_k \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{M}_k)^T \mathbf{C}_k^{-1}(\mathbf{x} - \mathbf{M}_k)\right). \quad (5)$$

Values of D , G , $(wg_k)_{i=1}^G$, $(\mathbf{M}_k)_{k=1}^G$, $(\mathbf{C}_k^{-1})_{k=1}^G$ describe the estimator of PDF.

3 Monte Carlo Integration Method – The Theoretical Background

Obtaining a selectivity value given by (1) is equivalent to the estimation of a multiple integral value and it will be based on the well-known Monte-Carlo method and the mean-value theorem of calculus.

Let us assume:

$\mathbf{a} = [a_1, \dots, a_D]^T$ and $\mathbf{b} = [b_1, \dots, b_D]^T$ – vectors of the range query bounds, $[a_1, b_1] \times \dots \times [a_D, b_D]$ – query condition hyper-rectangle, denoted by $QCHR$, V – integration volume (volume of $QCHR$):

$$V = \prod_{i=1}^D (b_i - a_i) \quad (6)$$

$\mathbf{x}_j = [x_{1j}, \dots, x_{ij}, \dots, x_{Dj}]^T$ – random single vector uniformly generated from $QCHR$ for $j = 1 \dots N$,
 $f_j = f(\mathbf{x}_j)$ – value of PDF for \mathbf{x}_j sample,
 I – required value of the multi integral (see eq. 1):

$$I = \int_{a_1}^{b_1} \dots \int_{a_D}^{b_D} f(x_1, \dots, x_D) dx_1 \dots dx_D = \int_{x \in QCHR} f(\mathbf{x}) d\mathbf{x} \quad (7)$$

$\langle f \rangle$ – mean value of $f(x)$ over $QCHR$:

$$\langle f \rangle = \frac{1}{V} \int_{x \in QCHR} f(\mathbf{x}) d\mathbf{x}. \quad (8)$$

Using (7) and (8) we may obtain:

$$I = V \langle f \rangle. \quad (9)$$

Let us assume:

$\widehat{\langle f \rangle}$ – estimator of $\langle f \rangle$:

$$\widehat{\langle f \rangle} = \frac{1}{N} \sum_{j=1}^N f_j \quad (10)$$

\widehat{I} – estimator of I (the selectivity estimator) based on N samples:

$$\widehat{I} = V \widehat{\langle f \rangle} = \frac{V}{N} \sum_{j=1}^N f_j \quad (11)$$

$\langle f^2 \rangle$ – mean value of $f^2(x)$,

$\widehat{\langle f^2 \rangle}$ – estimator of $\langle f^2 \rangle$ based on N samples:

$$\widehat{\langle f^2 \rangle} = \frac{1}{N} \sum_{j=1}^N f_j^2 \quad (12)$$

s_f^2 – variance of $f(x)$,

$$s_f^2 = \langle f^2 \rangle - \langle f \rangle^2 \quad (13)$$

\widehat{s}_f^2 – estimator of variance of $f(x)$ based on N samples.

Using central limit theorem we may find the estimator of the variance of $\langle f \rangle$:

$$\widehat{s}^2 = \frac{\widehat{s}_f^2}{N} = \frac{\widehat{\langle f^2 \rangle} - (\widehat{\langle f \rangle})^2}{N} \quad (14)$$

and the estimator of standard deviation of $\langle f \rangle$:

$$\widehat{s} = \sqrt{\frac{\widehat{s}_f^2}{N}} = \frac{\widehat{s}_f}{\sqrt{N}} = \frac{\sqrt{\widehat{\langle f^2 \rangle} - (\widehat{\langle f \rangle})^2}}{\sqrt{N}}. \quad (15)$$

Using (9) and (15) we may find the estimator of standard deviation of I :

$$\widehat{s}_I = V\widehat{s} = V\frac{\widehat{s}_f}{\sqrt{N}} = V\frac{\sqrt{\langle \widehat{f}^2 \rangle - (\langle \widehat{f} \rangle)^2}}{\sqrt{N}}. \quad (16)$$

Finally, we may obtain the approximate value of selectivity and the estimation of selectivity approximation error:

$$\widehat{I} \pm \alpha \widehat{s}_I = \widehat{I} \pm \alpha V\frac{\widehat{s}_f}{\sqrt{N}} = V\langle \widehat{f} \rangle \pm \alpha V\frac{\sqrt{\langle \widehat{f}^2 \rangle - (\langle \widehat{f} \rangle)^2}}{\sqrt{N}} \quad (17)$$

where α is a parameter needed for setting the confidence interval for I , i.e. $[\widehat{I} - \alpha \widehat{s}_I, \widehat{I} + \alpha \widehat{s}_I]$. Because I has a normal distribution, we may use the three sigma rule for setting the size of confidence interval, i.e. we may choose α . For $\alpha = 1, 2, 3$ the confidence levels (probabilities that I values belong to relevant confidence intervals) equal about 0.65, 0.95, 0.997, respectively.

The estimation error is given by:

$$E(N) = \alpha V\frac{\widehat{s}_f}{\sqrt{N}}. \quad (18)$$

Using big- \mathbf{O} notation for (18), we can say that formula $\mathbf{O}(N^{-\frac{1}{2}})$ describes the dependency between the estimation of an error value and the number of samples for MC method. There is no dependency on D .

For $D \gg 6$ MC method becomes better than standard methods of integration (e.g. Trapezoidal, Simpson etc.) when N is increased. For example, the error estimation for the method based on 1-dimensional Trapezoidal rule ([5] chapt. 11) is $\mathbf{O}(h^2) = \mathbf{O}(N^{-2})$. For D -dimensional Trapezoidal rule it is $\mathbf{O}(N^{-\frac{2}{D}})$, thus MC becomes better ($N^{-\frac{1}{2}} < N^{-\frac{2}{D}}$) for $D > 4$. This is the advantage of MC method for high dimensions.

The method of obtaining the result integral estimator with given error E_M (the maximal absolute estimation error) consists of two stages: the work step stage and the normal step one.

At the work step stage, we use $N_{WorkStep}$ – some small number of samples (typically $N_{WorkStep} = 1000$). First we calculate $\widehat{s}_f(N_{WorkStep})$. Using the following inequality:

$$E(N) = \frac{\alpha V \widehat{s}_f(N_{WorkStep})}{\sqrt{N}} < E_M \quad (19)$$

we may find N_{Min} – the estimated number of samples needed to ensure that error is less than E_M :

$$N \geq N_{Min} = \text{ceil} \left(\frac{\alpha V \widehat{s}_f(N_{WorkStep})}{E_M} \right)^2. \quad (20)$$

At the normal step stage, we calculate $\widehat{I}(N_{Min})$, $\widehat{s}_f(N_{Min})$, $E(N_{Min})$ using additional $N_{FinalStep} = N_{Min} - N_{WorkStep}$ samples. There is a chance that the

condition $E(N_{Min}) < E_M$ is not satisfied, so the next normal step is needed etc. In practice, only one final step is enough (in most cases, the error condition is satisfied after taking into account N_{Min} samples).

4 GPU-Accelerated Selectivity Estimation

To take advantage of massive parallelism available on GPU, the MC method described in the previous section was adopted to GPU specificity and implemented in CUDA environment [10] with C language.

On the GPU side all computations are done within a kernel function. As an input it takes: the number of samples to process (N), the number of dimensions (D), the number of Gaussian clusters used (G), the number of samples processed by each thread ($TWS - ThreadWorkSize$), integration bounds (\mathbf{a} and \mathbf{b} vectors), modified weights ($(wg_k)_{k=1}^G$), clusters centers ($(\mathbf{M}_k)_{k=1}^G$) and inverted covariance matrices ($(\mathbf{C}_k^{-1})_{k=1}^G$). As a result, it returns two vectors sum_f and sum_f2 , containing sums needed to calculate \widehat{f} (see eq. 10) and $\widehat{f^2}$ (see eq. 12), respectively. These are partial results (calculated in each thread block), aggregated later to final result on the CPU side. The kernel function uses random number generators from CURAND Library [9].

Computations made on the GPU side are done in a 1-dimensional thread blocks, which have a size equals $BLOCK_SIZE = 2 \times WARP_SIZE$ (which is 64 in most cases). Blocks are grouped within a 1-dimensional grid, which size is calculated according to the following formula: $ceil(\frac{N}{TWS \times BLOCK_SIZE})$.

To reduce the total computation time, several well known optimization techniques were used. To minimize time spent on accessing GPU global memory, threads within a block use shared memory, where $(wg_k)_{k=1}^G$, $(\mathbf{M}_k)_{k=1}^G$, $(\mathbf{C}_k^{-1})_{k=1}^G$, \mathbf{a} , \mathbf{b} are stored. Also the reduction mechanism [7] was utilized when partial sums in a block are calculated. Here, we used warp-synchronous approach to avoid an explicit thread synchronization, hence $BLOCK_SIZE = 2 \times WARP_SIZE$.

All computations on GPU are done with a single precision.

4.1 Tuning Process of Obtaining Time-Optimal Parameters

For our algorithm of selectivity calculation we introduced parameters: $T_{MaxWorkStep}$ (maximum time for a work step) and $T_{AllSteps}$ (maximum time for a whole selectivity calculation). We used $T_{MaxWorkStep} = 0.25ms$ and $T_{AllSteps} = 1ms$, respectively.

During the update statistics we obtain not only $(wg_k)_{k=1}^G$, $(\mathbf{M}_k)_{k=1}^G$, $(\mathbf{C}_k^{-1})_{k=1}^G$ but also time-optimal parameters for selectivity calculation. TWS is one of the most important because it affects the performance. It depends on D and G , but does not depend on the other parameters of the PDF estimator.

For the work step stage with given D , G , $N = N_{WorkStep} = 1000$ and the time constraint $T_{WorkStep} \leq T_{MaxWorkStep}$, we obtain time-optimal TWS and the corresponding value of $T_{WorkStep}$ according to the Alg. 1. We also obtain the new improved $N_{WorkStep}$ value, which does not have to be exact 1000 i.e.

Algorithm 1. Obtaining time-optimal parameters for work step stage

```

procedure TUNEPARAMSWORK( $N, D, G, (wg_k)_{k=1}^G, (M_k)_{k=1}^G, (C_k^{-1})_{k=1}^G$ )
   $T_{WorkStep} \leftarrow 0.25ms$ 
  for  $tws = 1 \rightarrow MAX\_TWS$  do  $\triangleright MAX\_TWS = 20$ 
     $t_1 \leftarrow time()$ 
     $sel, N' \leftarrow CalcSelectivity(tws, N, D, G, (wg_k)_{k=1}^G, (M_k)_{k=1}^G, (C_k^{-1})_{k=1}^G)$ 
     $t_2 \leftarrow time()$ 
    if  $t_2 - t_1 \leq T_{WorkStep}$  then
       $TWS_{WorkStep} \leftarrow tws, T_{WorkStep} \leftarrow t_2 - t_1, N_{WorkStep} \leftarrow N'$ 
    end if
  end for
  return  $TWS_{WorkStep}, T_{WorkStep}, N_{WorkStep}$ 
end procedure

```

it may be a little greater, if it has no influence on increasing $T_{WorkStep}$ (see N' in the Alg. 1) . Finally, for a given D, G we have the optimal: $TWS_{WorkStep}, T_{WorkStep}, N_{WorkStep}$.

We assume that there is only one final step. For the normal step stage with given D, G , and the time constraint: $T_{FinalStep} \leq T_{AllSteps} - T_{WorkStep}$, we obtain, according to the Alg. 2, the time-optimal TWS and the maximum number of samples processed during $T_{FinalStep}$ i.e. $N_{FinalStep}$. Finally, for a given D, G we have the optimal: $TWS_{FinalStep}, T_{FinalStep}, N_{FinalStep}$.

Algorithm 2. Obtaining time-optimal parameters for normal step stage

```

procedure TUNEPARAMSNORMAL( $D, G, (wg_k)_{k=1}^G, (M_k)_{k=1}^G, (C_k^{-1})_{k=1}^G$ )
   $N_{FinalStep} \leftarrow N_{WorkStep}, T_{FinalStep} \leftarrow 1ms - T_{WorkStep}, break \leftarrow false$ 
  repeat
     $N_{tmp} \leftarrow N_{FinalStep} + \Delta N$   $\triangleright \Delta N = 100$ 
     $tws, t, N_{tmp} \leftarrow TuneParamsWork(N_{tmp}, D, G, \dots)$ 
    if  $t \leq T_{FinalStep}$  then
       $TWS_{FinalStep} \leftarrow tws, T_{FinalStep} \leftarrow t, N_{FinalStep} \leftarrow N_{tmp}$ 
    else
       $break \leftarrow true$ 
    end if
  until  $\neg break$ 
  return  $TWS_{FinalStep}, T_{FinalStep}, N_{FinalStep}$ 
end procedure

```

4.2 CUDA-Based Selectivity Estimation Method

The selectivity estimation method (invoked for concrete **a** and **b**) utilizes meta-data (TWS, T, N) that were obtained for the work step and the final one during update statistics.

At the beginning, the work step is performed with $TWS_{WorkStep}$, $N_{WorkStep}$. If $N_{Min} - N_{WorkStep} \leq 0$ (case 1), the work step is sufficient. CPU finishes the calculations and the resulting integral value $\hat{I}(N_{WorkStep})$, and $E(N_{WorkStep})$ value (less than E_M) are returned.

If $N_{Min} - N_{WorkStep} \leq N_{FinalStep}$ (case 2), the final step is performed with $TWS_{FinalStep}$, $N_{FinalStep}$. CPU finishes the calculations and the resulting integral value $\hat{I}(N_{WorkStep} + N_{FinalStep})$, and $E(N_{WorkStep} + N_{FinalStep})$ value (less than E_M) are returned.

If $N_{Min} - N_{WorkStep} > N_{FinalStep}$ (case 3), the final step is not sufficient. We may return $E(N_{WorkStep} + N_{FinalStep})$ before the optional final step execution. This allows CQO (which invokes this algorithm) to decide if the expected worse accuracy is acceptable and the final step should be performed or the optimizer will use a completely different method of selectivity estimation.

5 Selected Experimental Results

Experiments were conducted on a low-budget GPU device NVIDIA Quadro FX 580 and CPU Intel Xenon W3550 @ 3.07 GHz. We measured a time needed to transfer \mathbf{a} and \mathbf{b} from CPU to GPU, execute GPU kernel, transfer sum_f and sum_f2 from GPU to CPU, and finally, sum partial results and calculate resulting selectivity, error estimator and N_{Min} estimator. We do not take into account transferring PDF parameters from CPU to GPU and initialization of random generators. These activities may be done once and there is no need to repeat them with every invoke of selectivity method.

Here, we will present the selected experimental results performed for $D = 4$, $G = 6$. Tab. 1 presents results of a parameters tuning process for a work step obtained with Alg. 1 (some of them were omitted for clarity). The shortest calculation time ($\sim 0.22ms$) is for $TWS_{WorkStep} = 2$.

Table 1. Obtaining the time-optimal parameter values for the work and final steps

$TWS_{WorkStep}$	$T_{WorkStep}$ [ms]	$N_{WorkStep}$	$TWS_{FinalStep}$	$T_{FinalStep}$ [ms]	$N_{FinalStep}$
1	0.2318	1024	5	0.4719	11200
2	0.2155	1024	20	0.6110	16640
3	0.2313	1152	9	0.6346	17280
4	0.2410	1024	9	0.6850	18432
5	0.2563	1280	10	0.7426	21120

Then we can find the optimal values of $TWS_{FinalStep}$ and $N_{FinalStep}$ parameters assuming that computations will take no longer then $0.78ms$ ($T_{WorkStep} + T_{FinalStep} = T_{AllSteps} = 1ms$). This is done according to the Alg. 2. Tab. 1 presents results obtained with this algorithm (some of them were omitted for clarity). In this particular example: $TWS_{FinalStep} = 10$, $N_{FinalStep} = 21120$ and overall selectivity calculation time should be about $0.96ms$.

Here, we will present the example selectivity estimation process which is based on metadata obtained in the described-above tuning process. We assume $\alpha = 2$, $E_M = 0.01$ and the following parameters of the PDF estimator: $(w_k)_{k=1}^6 = [0.12 \ 0.3 \ 0.19 \ 0.18 \ 0.11 \ 0.1]$,

$$(\mathbf{M}_k)_{k=1}^6 = \begin{bmatrix} 0.18 & 0.2 & 0.5 & 0.11 & 0.75 & 0.85 \\ 0.25 & 0.3 & 0.5 & 0.14 & 0.76 & 0.86 \\ 0.23 & 0.4 & 0.4 & 0.13 & 0.77 & 0.87 \\ 0.24 & 0.2 & 0.5 & 0.12 & 0.78 & 0.88 \end{bmatrix}, \quad (21)$$

$\mathbf{C}_k = \text{diag}(0.01, 0.01, 0.01, 0.01)$ for $k = 1 \dots 6$.

Let us consider the query condition where $a_i = 0.3$ and $b_i = 0.6$ for $i = 1 \dots 4$ (the same query boundaries in all dimensions). During the work step (run with $N_{WorkStep} = 1024$ and $TW_{WorkStep} = 2$) we obtain results: $\hat{I}(1024) = 0.0896$, $E(1024) = 0.0049$, and $N_{Min} = 247 < 1024$. This means that the case 1 occurred and the process of selectivity calculation is finished (during about $T_{WorkStep} = 0.2154ms$).

Let us consider the query condition with wider ranges i.e. where $a_i = 0.125$ and $b_i = 0.65$ for $i = 1 \dots 4$. During the work step (with the same $N_{WorkStep}$ and $TW_{WorkStep}$ as above) we obtain: $\hat{I}(1024) = 0.3884$, $E(1024) = 0.046$, and $N_{Min} = 21658 \in (1024, 1024 + 21120]$. This situation was denoted as the case 2 and after the final step performed with $N_{WorkStep} = 21120$ and $TW_{WorkStep} = 10$ we obtain: $\hat{I}(1024 + 21120) = 0.3757$, $E(1024 + 21120) = 0.0094$. Both steps took about $1ms$.

We may compare this to the execution time of the referential single-threaded CPU-based module for selectivity estimation. Selectivity calculating for the same query condition (case 2) took about $16ms$ and it does not satisfy the assumed time constraint.

The situation when performing the final step is not enough (case 3) is possible of course, but it is rather unlikely to occur. For example if we assume that distributions of pairs (a_i, b_i) for $i = 1 \dots D$ ($D = 4$) are independent and described by the following 2D-uniform PDF:

$$f_{truncated2D-uniform}(a_i, b_i) = \begin{cases} 2 & \text{for } 0 \leq a_i \leq 1 \wedge 0 \leq b_i \leq 1 \wedge a_i \leq b_i \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

we checked that the case 3 ($N_{Min} - N_{WorkStep} > N_{FinalStep}$, i.e. $N_{Min} > 1024 + 21120$) occurs only for about 0.6% of query conditions. This means that selectivities for a significant number of query conditions are obtained with the error less than E_M . About 92.2% of query conditions are handled during the work step and about 7.4% during the final one.

6 Conclusions

The paper describes the Monte Carlo-based method of complex range query selectivity estimation which was adopted to parallel GPU processing. Applying

GPU allows to use MC approach to time-critical on-line selectivity calculation, what is impossible in most cases, when a classic single-threaded CPU-based solution is used because of execution time constraints. The paper also describes algorithms for tuning the method parameters.

Future plans will concentrate on further improvements of the method, e.g. by using quasi-random sequences for variance reduction. Especially, we will consider using Sobol's sequences that are also supported by CURAND Library.

Acknowledgments. This work was supported by the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09-02).

References

1. Augustyn, D.R., Zederowski, S.: Applying cuda technology in dct-based method of query selectivity estimation. In: ADBIS Workshops, pp. 3–12 (2012)
2. Böhm, C., Kriegel, H.-P., Kröger, P., Linhart, P.: Selectivity estimation of high dimensional window queries via clustering. In: Medeiros, C.B., Egenhofer, M., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 1–18. Springer, Heidelberg (2005)
3. Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K.: Approximate query processing using wavelets. *The VLDB Journal* 10, 199–223 (2001)
4. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. *SIGMOD Rec.* 30, 461–472 (2001)
5. Gould, H., Tobochnik, J., Wolfgang, C.: *An Introduction to Computer Simulation Methods: Applications to Physical Systems*, 3rd edn. Addison-Wesley Longman Publishing Co., Inc., Boston (2005)
6. Gustafson, D.E., Kessel, W.C.: Fuzzy clustering with a fuzzy covariance matrix. In: *Proc. IEEE CDC*, San Diego, CA, USA, pp. 761–776 (1979)
7. Harris, M.: *Optimizing Parallel Reduction in CUDA* (2011), <http://developer.download.nvidia.com/assets/cuda/files/reduction.pdf>
8. Khachatryan, A., Müller, E., Böhm, K., Kopper, J.: Efficient selectivity estimation by histogram construction based on subspace clustering. In: Bayard Cushing, J., French, J., Bowers, S. (eds.) *SSDBM 2011*. LNCS, vol. 6809, pp. 351–368. Springer, Heidelberg (2011)
9. NVidia Corporation: *CUDA Toolkit 5.0 CURAND Guide*, version 5.0 (2012), http://docs.nvidia.com/cuda/pdf/CURAND_Library.pdf
10. NVidia Corporation: *NVIDIA CUDA™C Programming Guide*, version 5.0 (2012), <http://docs.nvidia.com/cuda/pdf/CUDA-C-Programming-Guide.pdf>
11. Scott, D.W., Sain, S.R.: *Multi-Dimensional Density Estimation*, pp. 229–263. Elsevier, Amsterdam (2004)
12. 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)

Exploring the Design Space of a GPU-Aware Database Architecture

Sebastian Breß¹, Max Heibel², Norbert Siegmund¹,
Ladjel Bellatreche³, and Gunter Saake¹

¹ School of Computer Science
University of Magdeburg

{sebastian.bress,nsiegmun,gunter.saake}@ovgu.de

² Technische Universität Berlin
max.heibel@tu-berlin.de

³ LIAS/ISAE-ENSMA, Futuroscope, France
bellatreche@ensma.fr

Abstract. The vast amount of processing power and memory bandwidth provided by modern graphics cards make them an interesting platform for data-intensive applications. Unsurprisingly, the database research community has identified GPUs as effective co-processors for data processing several years ago. In the past years, there were many approaches to make use of GPUs at different levels of a database system. In this paper, we summarize the major findings of the literature on GPU-accelerated data processing. Based on this survey, we present key properties, important trade-offs and typical challenges of GPU-aware database architectures, and identify major open research questions.

1 Introduction

Over the last ten years, *Graphics Processing Units* (GPUs) matured from highly specialized processing elements to fully programmable, powerful co-processors. This development has inspired the database research community to investigate methods for accelerating database systems via GPU co-processing. Several research papers and performance studies demonstrate the potential of this approach [3,9,14,24] – and the technology has also found its way into commercial products (e.g., Jexod [1]).

Using graphics cards to accelerate data processing is tricky and has several pitfalls: First, for effective GPU co-processing, the data transfer bottleneck between CPU and GPU has to either be reduced or be concealed via clever data placement or caching strategies. This is a challenging task for existing systems, given that CPU and GPU often use vastly different data representations. Second, when integrating GPU co-processing into a “real-world” *Database Management System* (DBMS), we have to overcome the problem that DBMS internals, such as data structures, query processing and optimization, are optimized for CPUs. While there is ongoing research on building GPU-aware database systems [7,10], so far no unified GPU-aware DBMS architecture has emerged.

In this paper, we want to make the community aware of the lack of a GPU-aware architecture and derive – based on a literature survey – a reduced design space of such an architecture. In particular, we make the following contributions: (1) We traverse the design space for a GPU-aware database architecture with respect to functional and non-functional properties based on results of prior work, and (2) derive research questions that should be investigated by the community.

The paper is structured as follows. In Section 2, we provide necessary background information about GPUs. We explore the design space for GPU-accelerated DBMS w.r.t. functional and non-functional properties in Section 3. Finally, we summarize our findings and identify open challenges and research questions.

2 Background: Graphics Card Architecture

Figure 1 showcases¹ the architecture of a modern computer system with a graphics card. The graphics card – henceforth also called the *device* – is connected to the *host system* via the *PCIExpress bus*. All data transfer between host and device has to pass through this comparably low-bandwidth bus.

The graphics card contains the GPU and a few² gigabytes of *device memory*. Typically, host and device do not share the same address space, meaning that neither the GPU can directly access the main memory nor the CPU can directly access the device memory.

The GPU itself consists of a few *multiprocessors*, which can be seen as very wide SIMD processing elements. Each multiprocessor packages several *scalar processors* with a few kilobytes of high-bandwidth, on-chip *shared memory* and an interface to the device memory.

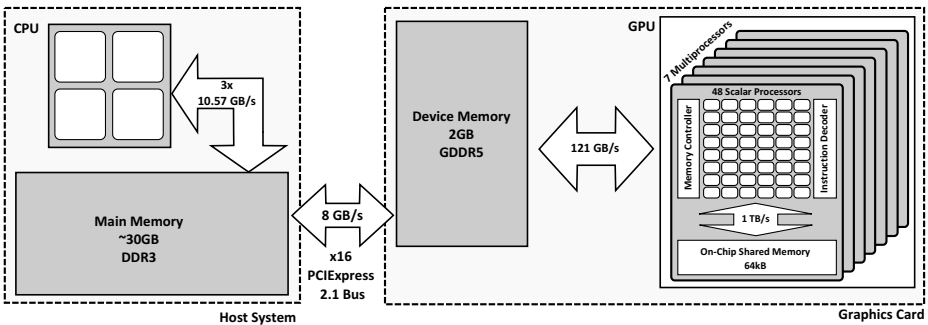


Fig. 1. Overview: Exemplary architecture of a system with a graphics card

¹ The figure shows the architecture of a graphics card from the *Fermi* architecture of NVIDIA. While specific details might be different for other vendors, the general concepts are found in all modern graphic cards.

² Typically around 2GB on mainstream cards and up to 16GB on high-end devices.

3 Exploring the Design Space of a GPU-Aware DBMS Architecture

In this section, we explore the design space of a GPU-accelerated database management system. We will look at the design space from two point of views: non-functional properties (e.g., performance and portability) and functional properties (e.g., transaction management and processing model). Note that while we focus on relational systems, most of our discussions apply to NoSQL and multi-dimensional DBMS as well.

3.1 Non-functional Properties

In the following, we discuss non-functional properties that a DBMS may be optimized for and the upcoming problems in case GPU-acceleration should be supported.

Performance. Since the GPU is a specialized processor, it can do certain tasks (e.g., numeric computation) faster than CPUs, whereas CPUs outperform GPUs for tasks that are hard to parallelize or involve a lot of control flow instructions in the algorithm. He and others observed that joins are 2–7 times faster on the GPU, whereas selections are 2–4 times slower in case data transfers are needed [15]. Gregg and Hazelwood showed that a GPU algorithm is not necessarily faster than its CPU counterpart [13], which is mainly caused by expensive data transfers between CPU and GPU RAM.

We argue that it is therefore a non-trivial problem to select the correct processing device for an operation in a query plan. We identify four major factors that need to be considered in such a decision [7]: (1) the operation to execute, (2) the features of the input data (e.g., data size, data type, operation selectivity, data skew), (3) the computational power and capabilities of the processing devices (e.g., number of cores, memory bandwidth, clock rate), and (4) the load on the processing device (e.g., even if an operation is typically faster on the GPU, one should use the CPU when the GPU is overloaded).

Through some operations are typically faster (e.g., aggregations) or slower (e.g., selections) on the GPU, using rules of thumb is unlikely to achieve good performance because of the large parameter space. Therefore, we argue that one needs a decision model for operator placement (e.g., Breß and others [5] or He and others [14]).

Portability. DBMS should be portable over a large variety of hardware. In case of a GPU-accelerated DBMS, this is a non-trivial task, because using vendor-specific toolkits such as CUDA binds the DBMS to a certain GPU vendor. There are two possibilities to counter this: First, implementing all operators for all vendor-specific toolkits. While this has the advantage that special features of a vendor's product can be used to achieve high performance, it leads to high implementation effort and development cost. Examples for such systems are GDB

[14] or CoGaDB, a column-oriented and gpu-accelerated DBMS.³ Second, implementing the operators in a generic framework, such as OpenCL, and letting the hardware vendor provide the optimal mapping to the given GPU. While this approach saves implementation efforts and simplifies maintenance, it likely suffers from performance degradation when compared to vendor-specific frameworks. To the best of our knowledge, the only system belonging to the second class is Ocelot [18], which extends MonetDB with OpenCL-based operators.

3.2 Functional Properties

We now discuss the design space for a GPU-accelerated DBMS with respect to functional properties in consideration of research results in this field. For this, we will take a look at the following design decisions: (1) main-memory vs. disk-based system, (2) row-oriented vs. column-oriented storage, (3) processing models (tuple-at-a-time model vs. operator-at-a-time), (4) GPU-only vs. hybrid device database, (5) GPU buffer management (column-wise or page-wise buffer management?), (6) query optimization for hybrid systems, and (7) consistency and transaction processing (lock-based vs. lock free protocols).

Main-Memory vs. Hard-Disk-Based System. He and others demonstrated that GPU-acceleration cannot achieve significant speedups if the data has to be fetched from disk, because of the IO bottleneck, which dominates execution costs [14]. Since the GPU only improves performance once the data has arrived in main memory, time savings will be small compared to the total query runtime. Hence, a GPU-aware database architecture should make heavy use of in-memory technology.

Row-Stores vs. Column Stores. Ghodsnia compares row and column stores with respect to their suitability for GPU-accelerated query processing [11]. Ghodsnia concluded that a column store is more suitable than a row store, because a column store (1) allows for coalesced memory access on the GPU, (2) achieves higher compression rates, an important property considering the current memory limitations of GPUs, and (3) reduces the volume of data that needs to be transferred. For example, in case of a column store, only those columns needed for data processing have to be transferred between processing devices. In contrast, in a row-store, either the full relation has to be transferred or a projection has to reduce the relation to the data needed to process a query. Both approaches are more expensive than storing the data column wise. Bakkum and others came to the same conclusion [2]. Furthermore, given that we already concluded that a GPU-aware DBMS should be an in-memory database system, and that current research provides an overwhelming evidence in favor of columnar storage for in-memory systems [4], we conclude that a GPU-aware DBMS should use columnar storage.

³ http://www.witi.cs.uni-magdeburg.de/iti_db/research/gpu/cogadb/

Processing Model. There are basically two alternative processing models that are used in modern DBMS: the tuple-at-a-time volcano model [12] and operator-at-a-time bulk processing [21]. Tuple-at-a-time processing has the advantage that intermediate results are very small, but has the disadvantage that it introduces a higher per tuple processing overhead as well as a high miss rate in the instruction cache. In contrast, operator-at-a-time processing leads to cache friendly memory access patterns, making effective usage of the memory hierarchy. The major drawback is the increased memory requirement, since intermediate results are materialized [21].

In the context of GPU-accelerated data management, operator-at-a-time processing is more promising than tuple-at-a-time processing, because data can be most efficiently transferred over the PCIe bus by using large memory chunks [22]. Therefore, approaches such as the tuple-at-a-time processing [12] may exhibit a poor performance, because they lead to underutilization of the PCIe bus. Furthermore, we identified in prior work that tuple-wise processing is not possible on the GPU, because inter-kernel communication is undefined [8]. Using the operator-at-a-time processing scheme avoids this problem. A further advantage is that the operator-at-a-time processing can be easily combined with operator-wise scheduling.

Database in GPU RAM vs. Hybrid Device Database. Ghodsnia proposed keeping the complete database resident in GPU RAM [11]. This approach has the advantage of vastly reducing data transfers between host and device. Also, since the GPU RAM has a roughly 25 times higher bandwidth than the PCIe Bus (2.0), this approach is very likely to increase performance significantly. It also simplifies transaction management, since data does not need to be kept consistent between CPU and GPU.

However, the approach has some obvious shortcomings: First, the GPU RAM (up to ≈ 16 GB) is rather limited compared to CPU RAM (up to ≈ 2 TB). This limits the approach to comparably small data sets, and forces the system to partition data across multiple GPUs to allow reasonable data sizes. This complicates processing and limits the number of applications that can make use of the system. Second, a pure GPU database cannot exploit full inter-device parallelism, because the CPU does not perform data processing. Since CPU and GPU both have their corresponding sweet-spots for different applications (cf. 3.1), this is a major shortcoming that significantly limits performance in several scenarios.

Since the problems outweigh the benefits, we conclude that a GPU-aware DBMS should make use of all available processing devices and not limit itself to GPU RAM. While this complicates data processing, and requires a data-placement strategy⁴, we still expect the hybrid to be faster than a pure CPU or GPU-based system.

⁴ Some potential strategies include keeping the hot set of the data resident on the graphics card, or using the limited graphics card memory as a low-resolution data storage to quickly filter out non-matching data items [23].

Effective GPU Buffer Management. The buffer management problem in a CPU/GPU system is similar to the one encountered in “traditional” disk-based or in-memory systems: We want to process data in a faster, and smaller memory space (GPU RAM), where the data is stored in a larger and slower memory space (CPU RAM). The novelty in this problem is, that – in contrast to “traditional” systems – data can be processed in both memory spaces. In other words: We can transfer data, but we don’t have to. This “optionality” opens up some interesting research questions, that have not been covered so far.

Data structures and data encoding are often highly optimized for the special properties of a processing device to maximize performance. Hence, different kinds of processing devices use an encoding optimized for the respective device (e.g., a CPU encoding has to support effective caching to reduce the memory access cost [20], whereas a GPU encoding has to ensure coalesced memory access of threads to achieve maximal performance [22]). This usually requires trans-coding data before or after the data transfer, which is additional overhead that can break performance.

Another interesting design decision is the granularity that should be used for managing the GPU RAM: pages, whole columns, or whole tables? Since we already concluded that a GPU-accelerated database should be columnar, this basically boils down to comparing page-wise vs. column-based caching. Page-wise caching has the advantage that it is an established approach, and is used by almost every DBMS, which eases integration into existing systems. However, a possible disadvantage is that depending on the page size, the PCIe bus may be underused during data transfers. Since it is more efficient to transfer few large data sets than many little datasets (with the same total data volume) [22], it could be more beneficial to cache and manage whole columns.

Query Placement and Optimization. Given that a GPU-aware DBMS has to manage multiple processing devices, a major problem is to automatically decide which parts of the query should be executed on which device. This highly depends on multiple factors, including the operation, size and shape of the input data, processing power and computational characteristics of CPU and GPU as well as the optimization criterion. Optimizing for response time requires to split a query in parts, so that CPU and GPU can process a part of the query in parallel. For workloads that require a high throughput, different heuristics have to be developed. Furthermore, given that we can freely choose between multiple different processing devices with different energy characteristics, non-traditional optimization criteria like energy-consumption or cost-per-tuple are highly interesting in the scope of GPU-aware DBMS.

He and others were the first to address hybrid CPU/GPU query optimization [14]. They used a Selinger-style optimizer to create initial query plans and then used heuristics and an analytical cost-model to split a workload between CPU and GPU. In our previous work, we proposed a framework that can perform cost-based operation-wise scheduling and cost-based optimization of hybrid CPU/GPU query plans, which is designed to be used with operator-at-a-time

bulk processing [6]. Heimerl and others suggest using GPUs to accelerate query optimization instead of query processing. This approach could help to tackle the additional computational complexity of query optimization in a hybrid system [17]. It should be noted that there is some similarity to the problem of query optimization in the scope of distributed and federated DBMS [19]. However, we believe that query processing in distributed systems is too different from query processing in hybrid CPU/GPU systems to simply reuse these results. In particular:

1. In a distributed system, nodes are autonomous. This is in contrast to hybrid CPU/GPU systems, because the CPU explicitly commands co-processors what they have to do.
2. In a distributed system, there is no global state. However, we have a global state in hybrid CPU/GPU systems, because the CPU knows which co-processor performs a certain operation on a specific dataset.
3. The nodes in a distributed system are homogeneous, in contrast to highly heterogeneous processors in hybrid CPU/GPU systems.
4. The nodes in a distributed system are loosely coupled, meaning that a node may lose network connectivity to the other nodes or might crash. In a hybrid CPU/GPU system, nodes are tightly bound. That is, no network outages are possible due to a high bandwidth bus connection, and a GPU does not go down due to a local software error, rather the whole hybrid database system crashes.

We conclude that traditional approaches for a distributed system do not take into account specifics of hybrid CPU/GPU systems. Therefore, tailor-made co-processing approaches are likely to outperform approaches from distributed or federated query-processing.

Consistency and Transaction Processing. While keeping data consistent in a distributed database is a widely studied problem, research on transaction management on the GPU is almost non-existent. The only work we are aware of was done by He and others [16] and indicates, that a locking-based strategy significantly breaks the performance of GPUs [16]. They developed a lock-free protocol to ensure conflict serializability of parallel transactions on GPUs. However, to the best of our knowledge, there is no work that explicitly addresses transaction management in a GPU-aware DBMS. It is therefore to be investigated, how the performance characteristics of established protocols of distributed systems are compared to tailor-made transaction protocols.

Essentially, there are three ways of maintaining consistency between CPU and GPU: (1) Each data item could be kept strictly in one place. In this case, we would not require any replication management, and would have to solve a modified allocation problem. (2) Use established replication mechanisms, such as read one write all or primary copy. (3) Perform updates always on one processing device (e.g., the CPU) and periodically synchronize these changes the other devices.

4 Discussion and Future Directions

In this paper, we argue that GPU-aware (co-processor-accelerated) database architectures are the natural next step for in-memory database systems to further reduce the “memory wall” due to inter-processing-device parallelization. Taking a look at existing work, we explored the design space of such a system and – summarizing our findings – we argue that a GPU-aware database architecture should be an in-memory, column-oriented DBMS using the operator-at-a-time bulk processing model, a co-processor and data-locality-aware query optimizer, which distributes the workload on all available (co-)processors as well as an optimistic transaction protocol, such as the timestamp protocol.

However, based on existing research, we cannot answer all architectural design decisions. Therefore, we identify *open challenges* and *research questions*.

4.1 Open Challenges on GPU-Accelerated Data Management

We identify the following *open challenges*:

1. GPU-accelerated databases try to keep relational data cached on the device to avoid data transfer. Since device memory is limited, this is often only possible for a subset of the data. Deciding which part of the data should be offloaded to the GPU – finding a so called *data placement strategy* – is a difficult problem that currently remains unsolved.
2. Due to result transfer costs, operators that generate a large result set are often unfit for GPU-offloading. Since the result size of an operation is typically not known before execution, predicting whether a given operator will benefit from the GPU is a hard problem.
3. GPU-accelerated operators are of little use for disk-based database systems, where most time is spent on disk I/O. Since the GPU improves performance only once the data is in main memory, time savings will be small compared to the total query runtime. Furthermore, disk-resident databases are typically very large, making it harder to find an optimal data placement strategy.
4. Having the option of running operations on a GPU increases the complexity of query optimization: The plan search space is drastically larger and a cost function that compares runtimes across architectures is required. While there has been prior work in this direction [5,6,14], GPU-aware query optimization remains an open challenge.

4.2 Research Questions

We identify the following research questions, which should be investigated in future work:

1. How can GPU-acceleration be integrated in column stores, and – in particular – how should an efficient data-placement and query optimization strategy for a GPU-aware DBMS look like?

2. Which parts of a database engine should be hardware-conscious (fine-tuned to a particular architecture), and which parts should be hardware-oblivious (implemented in a general framework like OpenCL, that can be mapped to multiple architectures at runtime)?
3. How does the performance differ when comparing distributed-query-processing approaches with tailor-made approaches for hybrid CPU/GPU systems?
4. What is a suitable transaction protocol that ensures ACID properties over all (co-)processors?
5. Is it feasible to include GPU-acceleration in an existing DBMS by changing the architecture successively (e.g., Ocelot) or are the necessary changes on DBMS architecture and software so invasive and expensive that a rewrite from scratch is necessary (e.g., CoGaDB)?

We hope to tackle most of these questions in the course of our research projects Ocelot [18] and CoGaDB. Ocelot investigates the research questions from the point of view of a hardware-oblivious database engine (OpenCL) whereas CoGaDB takes the position of a hardware-sensitive database engine (CUDA). Furthermore, the systems implement the results of our discussions in this paper differently: While Ocelot includes GPU-acceleration in an existing DBMS (MonetDB) altering the architecture iteratively, CoGaDB is a complete rewrite that implements the architectural design advises from this paper.

Acknowledgements. The work of Siegmund is supported by the German ministry of education and science (BMBF), number 01IM10002B. We thank Tobias Lauer from Jedox AG and the anonymous reviewers for their helpful feedback.

References

1. Palo gpu accelerator. White Paper (2010)
2. Bakkum, P., Chakradhar, S.: Efficient data management for gpu databases (2012), <http://pbakkum.com/virginian/paper.pdf>
3. Bakkum, P., Skadron, K.: Accelerating sql database operations on a gpu with cuda. In: GPGPU, pp. 94–103. ACM (2010)
4. Boncz, P.A., Kersten, M.L., Manegold, S.: Breaking the memory wall in monetdb. *Commun. ACM* 51(12), 77–85 (2008)
5. Breß, S., Beier, F., Rauhe, H., Sattler, K.-U., Schallehn, E., Saake, G.: Efficient co-processor utilization in database query processing. *Information Systems* (2013), <http://dx.doi.org/10.1016/j.is.2013.05.004>
6. Breß, S., Geist, I., Schallehn, E., Mory, M., Saake, G.: A framework for cost based optimization of hybrid cpu/gpu query plans in database systems. *Control and Cybernetics* 41(4) (2012)
7. Breß, S., Mohammad, S., Schallehn, E.: Self-tuning distribution of db-operations on hybrid cpu/gpu platforms. In: GvD. CEUR-WS, pp. 89–94 (2012)
8. Breß, S., Schallehn, E., Geist, I.: Towards optimization of hybrid CPU/GPU query plans in database systems. In: Pechenizkiy, M., Wojciechowski, M. (eds.) *New Trends in Databases & Inform. AISC*, vol. 185, pp. 27–35. Springer, Heidelberg (2012)

9. Diamos, G., Wu, H., Lele, A., Wang, J., Yalamanchili, S.: Efficient relational algebra algorithms and data structures for gpu. Technical report, Center for Experimental Research in Computer Systems, CERS (2012)
10. Fang, R., He, B., Lu, M., Yang, K., Govindaraju, N.K., Luo, Q., Sander, P.V.: Gpuqp: query co-processing using graphics processors. In: SIGMOD, pp. 1061–1063. ACM (2007)
11. Ghodsnia, P.: An in-gpu-memory column-oriented database for processing analytical workloads. In: The VLDB PhD Workshop. VLDB Endowment (2012)
12. Graefe, G.: Encapsulation of parallelism in the volcano query processing system. In: SIGMOD, pp. 102–111. ACM (1990)
13. Gregg, C., Hazelwood, K.: Where is the data? why you cannot debate cpu vs. gpu performance without the answer. In: ISPASS, pp. 134–144. IEEE (2011)
14. He, B., Lu, M., Yang, K., Fang, R., Govindaraju, N.K., Luo, Q., Sander, P.V.: Relational query co-processing on graphics processors. *ACM Trans. Database Syst.* 34, 21:1–21:39 (2009)
15. He, B., Yang, K., Fang, R., Lu, M., Govindaraju, N., Luo, Q., Sander, P.: Relational joins on graphics processors. In: SIGMOD, pp. 511–524. ACM (2008)
16. He, B., Yu, J.X.: High-throughput transaction executions on graphics processors. *PVLDB* 4(5), 314–325 (2011)
17. Heimel, M., Markl, V.: A first step towards gpu-assisted query optimization. In: ADMS. VLDB Endowment (2012)
18. Heimel, M., Saecker, M., Pirk, H., Manegold, S., Markl, V.: Hardware-oblivious parallelism for in-memory column-stores. In: VLDB. VLDB Endowment (2013)
19. Kossmann, D.: The state of the art in distributed query processing. *ACM Computing Surveys* 32(4), 422–469 (2000)
20. Manegold, S., Boncz, P.A., Kersten, M.L.: Optimizing database architecture for the new bottleneck: Memory access. *The VLDB Journal* 9(3), 231–246 (2000)
21. Manegold, S., Kersten, M.L., Boncz, P.: Database architecture evolution: Mammals flourished long before dinosaurs became extinct. *PVLDB* 2(2), 1648–1653 (2009)
22. NVIDIA. Nvidia cuda c programming guide, pp. 30–34 (2012), http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (accessed February 16, 2013)
23. Pirk, H.: Efficient cross-device query processing. In: The VLDB PhD Workshop. VLDB Endowment (2012)
24. Pirk, H., Manegold, S., Kersten, M.: Accelerating foreign-key joins using asymmetric memory channels. In: ADMS, pp. 585–597. VLDB Endowment (2011)

Dynamic Compression Strategy for Time Series Database Using GPU*

Piotr Przymus¹ and Krzysztof Kaczmariski²

¹ Nicolaus Copernicus University, Poland
eror@umk.mat.pl

² Warsaw University of Technology, Poland
k.kaczmariski@mini.pw.edu.pl

Abstract. Nowadays, we can observe increasing interest in processing and exploration of time series. Growing volumes of data and needs of efficient processing pushed research in new directions. GPU devices combined with fast compression and decompression algorithms open new horizons for data intensive systems. In this paper we present improved cascaded compression mechanism for time series databases build on Big Table-like solution. We achieved extremely fast compression methods with good compression ratio.

Keywords: time series database, lightweight lossless compression, GPU, CUDA.

1 Introduction

Specialized time series databases play important role in industry storing monitoring data for analytical purposes. These systems are expected to process and store millions of data points per minute, 24 hours a day, seven days a week, generating terabytes of logs. Due to regression errors checking and early malfunction prediction these data must be kept with proper resolution including all details. Solutions like OpenTSDB [11], TempoDB [3] and others deal very well with these kind of tasks. Most of them work on a clone of Big Table approach from Google [5], a distributed hash table with mutual ability to write and read data in the same time.

Usually systems compress data before writing to a long-term storage. It is much more efficient to store data for some time in a memory or disk buffer and compress it before flushing to disk. This process is known as a table row rolling. Current systems like HBase [1], Casandra [6] and others offer compression optimization for entire column family. This kind of general purpose compression is not optimized for particular data being stored (i.e. various time series with different compression potential stored in one column family).

Similar problems appear in in-memory database systems. Solutions based on GPU processing (like ParStream [2]) tend to pack as many data into GPU devices global memory as possible. Efficient data compression method would significantly improve abilities of these systems. An average internet service with about 10 thousands of simultaneously working users may generate around 80GB of logs every day. After

* The project is funded by National Science Centre, decision DEC-2012/07/D/ST6/02483.

compression they could fit into two Nvidia Tesla devices where average query can be processed within seconds compared to minutes in case of standard systems.

In case of time series compression ratio could be improved by a method tuned to types of data including its variability, span, differences, etc. However, tuning time slows down compression and often cannot fit into time window available in real time monitoring systems. This paper describes a dynamic compression strategy planner for time series databases using GPU processors with reasonable processing time and compression ratios. What is even more important, the resulting compressed data block can be decompressed very quickly directly into the GPU memory additionally allowing for ultra fast query processing, what we discussed in our previous publication [12].

The main contribution of this work is:

- three new implementations of patched compression algorithms on GPU
- a new dynamic compression planner for lightweight compression methods
- categorization for compression methods and reduction of configuration space for optimal plan searching
- evaluation of the achieved results on real-life data

Section 2.1 presents a general view of the system, section 2.2 contains the main contribution of our work: the dynamically optimized compression system. Experimental runtime results are contained in section 3 while section 4 concludes.

1.1 Motivation and Related Work

Optimal data compression of time series is an interesting and widely analysed computational problem. Lossless methods often use some general purpose compression algorithms with several modifications according to knowledge gathered from data. On the other hand, lossy compression approximate data using, for instance, splines, piecewise linear approximation or extrema extraction [9]. For industrial monitoring, lossy compression cannot be used due to possible degradation of anomalies.

In case of lossless compression one can use common algorithms (ZIP, LZ0) which tend to consume lot of computation resources [4,15] or lightweight methods which are faster but not so effective. Our dynamic method attempts to combine properties of both approaches: is lossless but much faster than common algorithms, offers good compression ratios and may be computed incrementally. Also ability to decompress values directly into the processor shared memory should improve GPU memory bandwidth and enable it to be used in many data intensive applications.

An important challenge is to improve compression factor with an acceptable processing time in case of variable sampling periods. Interesting results in the field of lossless compression done on GPU were presented by Fang et al. [8]. Using a query planner it was possible to achieve significant improvement in overall query processing on GPU by reducing data transfer time from RAM to global device's memory space. The strategy applied in our work is based on statistics calculated from inserted data and used to find an optimal cascaded compression plan for the selected lightweight methods.

In a time series database we often observe data grouped into portions of very different characteristics. Optimal compression should be able to apply different compression

plans for different time series and different time periods. Comparing to [8] and [4] we can achieve better results by using dynamic compression planning methods with automated compression tuning upon processed time series data.

2 Dynamically Optimized Compression System

2.1 Time Series Database Architecture

General View A typical time series database consists of three layers: data insertion module, data storage and querying engine. Our compression mechanism touches all the layers working as a middle tier between the data storage and the rest of the system. In this work we shall focus on data compression mechanism assuming that decompression used by the query engine is an obvious opposite process.

2.2 Data Insertion

Data Collection. The data acquisition from ongoing measurements, industrial processes monitoring [10], scientific experiments [13], stock quotes or any other financial and business intelligence sources has got continuous characteristic. These discrete observations T are represented by pairs of a *timestamp* and a *numerical value* (t_i, v_i) with the following assumptions: *a*) number of data points (timestamps and their values) in one time series should not be limited; *b*) each time series should be identified by a name which is often called a *metric name*; *c*) each time series can be additionally marked with a set of *tags* describing measurement details which together with metric name uniquely identifies time series; *d*) observations may not be done in constant time intervals or some points may be missing, which is probable in case of many real life data.

Initial Buffering. Due to optimization purposes, data sent to the data storage should be ordered and buffered into portions, minimizing necessary disk operations but also minimizing the distributed storage nodes intercommunication. Buffering also prepares data to be compressed and stored optimally in an archive. Simplicity of data model imposed separated column families for compressed and raw data. Time series are separately compacted into larger records (by a metric name and tags) containing a specified period of time (e.g. 15 minutes, 2 hours, 24 hours – depending on the number of observations). This step directly predeceases dynamic compression which is described in the next section.

2.3 Compression Algorithms

Patched Lightweight Compression. The main drawback of many lightweight compression schemes is that they are prone to outliers in the data frame. For example, consider following data frame $\{1, 2, 3, 2, 2, 3, 1, 1, 64, 2, 3, 1, 1\}$, one could use the 2 bits fixed-length compression to encode the frame, but due to the outlier (value 64) we have to use 6-bit fixed-length compression or more computationally intensive 4-bit dictionary compression. Solution to the problem of outliers has been proposed in [15] as a

modification to three lightweight compression algorithms. The main idea was to store outliers as exceptions. Compressed block consists of two sections: the first keeps the compressed data and the second exceptions. Unused space for exceptions in the first section is used to hold the offset of the following exceptions in the data in order to create linked list, when there is no space to store the offset of the next exception, a *compulsive exception* is created [15]. For large blocks of data, the linked lists approach may fail because the exceptions may appear sparse thus generate a large number of compulsory exceptions. To minimise the problem various solutions have been proposed, such as reducing the frame size [15] or algorithms that do not generate compulsive exceptions [7,14]. The algorithms in this paper are based largely on those described by Yan [14]. In this version of the compression block is extended by two additional arrays - exceptions position and values. Decompression involves extracting data using the underlying decompression algorithm and then applying a patch (from exceptions values array) in the places specified by the exceptions positions. As exceptions are separated, data patching can be done in parallel. During compression, each thread manages two arrays for storing exception values and positions. After compression, each thread stores exceptions in the shared memory, similarly exceptions from shared memory are copied to the global memory. Patched version of algorithms are only selected if compression ratio improves. Otherwise non patched algorithms are used. Therefore complex exceptions treatment may be omitted speeding up the final compression.

SCALE. Converts float values to integer values by scaling. This solution can be used in case where values are stored with given precision. For example, CPU temperature 56.99 can be written as 5699. The scaling factor is stored in compression header.

DELTA. Stores the differences between successive data points in frame while the first value is stored in the compression header. Works well in case of sorted data, such as measurement times. For example, let us assume that every 5 minutes the CPU temperature is measured starting from 1367503614 to 1367506614 (Unix epoch timestamp notation), then this time range may be written as $\{300, \dots, 300\}$.

(Patched) Fixed-length Minimum Bit Encoding (PFL and FL). FL and PFL compression works by encoding each element in the input with the same number of bits thus deleting leading zeros at the most significant bits in the bit representation. The number of bits required for the encoding is stored in the compression header. The main advantage of the FL algorithm (and its variants) is the fact that compression and decompression are highly effective on GPU because these routines contain no branching-conditions, which decrease parallelism of SIMD operations. For best efficiency dedicated compression and decompression routines are prepared for every bit encoding length with unrolled loops and using only shift and mask operations. Our implementation does not limit minimum encoding length to size of byte (as in [8]). Instead each thread (de)compresses block of eight values, thus allowing encoding with smaller number of bits. For example, consider following data frame $\{1, 2, 3, 2, 2, 3, 1, 2, 3, 1, 1\}$, one could use the 2 bits fixed-length compression to encode the frame.

(Patched) Frame-Of-Reference (PFOR and FOR). Works similarly to FL and PFL, except before compression it transforms each value into an offset from the reference

value (for example smallest value) in compression block. Reference value is then stored in compression header. In this situation, we need exactly $\lceil \log_2(\max - \min + 1) \rceil$ bits to encode each value in the frame. For example, this is useful when storing measurement times, consider time range $\{1367503614, \dots, 1367506614\}$, then using FOR we only need $\lceil \log_2(1367506614 - 1367503614 + 1) \rceil = 12$ bits to store each value in this range (as opposed to 31 bits without this transformation).

(Patched) Dictionary (DICT and PDICT). DICT is suitable for data that have only a small number of distinct values. It uses a dictionary of distinct values. For compression and decompression purposes, dictionary is loaded into the shared memory. Binary search is used during compression to lookup values, then an index of value is used to encode. Decompression simply retrieves values at given index from dictionary. DICT writes indexes using byte-aligned types, for better compression a combination with other compression algorithm should be used. For example, consider data frame $\{0, 500, 1500, 100, 100, 1500000, 100, 15000\}$ using DICT only 1 byte is needed to store each value (even less if combined with other compression algorithm) in comparison to pure FL where more than 2 bytes would have been used.

Run-Length-Encoding (RLE) and Patched Constant (PCONST). RLE encodes values with a pair: value and run length, thus using two arrays to compress data. Consider following data frame $\{1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3\}$, then RLE would create two arrays: values $\{1, 2, 3\}$ and run length $\{5, 4, 3\}$. PCONST is a specialized version of RLE where almost whole data frame consist of one value with some exceptions. This may be reconstructed using: frame length, constant value and PATCH arrays. For example, let us assume that a measurement is done every five minutes with some exceptions, then delta is almost always constant and equals 300, any other value will be stored as exception.

2.4 Cascaded Compression Planer

Cascaded compression can significantly improve the compression ratio. However, there are two problems arising. First, there is a risk that cost of decompression will neglect benefits from lower transfer costs. Second problem is arising when searching for an optimal compression methods composition. Even relatively short plan of cascaded compressions (i.e. using 6 compositions out of 10 algorithms with repetitions) may generate a very large search space (in our example $\sum_{i=1}^6 10^i = 1,111,110$). Significant reductions must be done in order to achieve fast compression and best plan fitting in a reasonable time. We assumed that the time limit is set by corresponding CPU performance measured for one base compression step (see next section). Therefore in our method, the whole compression process including copying data to GPU, data statistics evaluation, optimal plan searching and final compression plan execution must be always faster than mentioned limit.

Stage One: Static Planner – Reduction of Plans Search Space. In the first static stage we determined acceptable transitions between compression algorithms which were divided into three categories: initial transformation, base compression, helper

compression. The complete compression schema is always composed of algorithms selected from these ordered categories with the following purposes:

1. **Transformation algorithms (SCALE, DELTA).** All algorithms in this section are optional but may be used together (if present must be applied in the given order). Goal: Improve properties of data storage and prepare for better compression.
2. **Base compression algorithms (PDICT, PFL, PFOR, RLE, PCONST).** Only one algorithm may be selected as the base algorithm. All algorithms in this section use two or more arrays. Some of them, may qualify for further compression using *Helper compression algorithms*.
3. **Helper compression algorithms (FOR, FL, DICT).** The algorithms used to compress selected arrays from the previous step. Each of the resulting arrays can be compressed with only one algorithm. In order to minimize the stages of decompression PATCH algorithms, which could create new arrays for compression, are excluded. The base algorithm used may limit algorithms in this section. For example, exceptions and values arrays in all PATCH algorithms may only be compressed with FL.

Composition of all sensible paths between algorithms in these three categories leaves only 32 suitable compression plans out of former one million. The longest possible cascaded compression plan may be composed of six steps.

Stage Two: Hints System – Possibility of Manual Tuning. Another reduction of possible compression plans generated in the first stage can be done manually by a user speeding up further plan choosing. Number and types of hints may vary in different situations. For example, in time series systems timestamps are always sorted and if we consider separated compression methods for timestamps and values we may find different and better plans for them. A hint indicating sorted input may suggest using DELTA before base algorithms. Additionally, for every metric additional features may be specified or even specific compression algorithm may be enforced. Currently supported hints are located in Table 1.

Table 1. A sample set of hints for a time series compression planner

Hints	Meaning
SCALE, (P)FL, RLE DELTA, (P)FOR (P)DICT, PCONST	Enforces a specific compression algorithm in the plan.
SORTED	Specify whether the data is sorted.
TIMESTAMP	Automatically added by system to timestamps. Sets SORTED to True and SCALE to False.
DATA	Automatically added by system to time series values. If not otherwise specified sets SORTED to False and SCALE to False.

Stage Three: Dynamic Statistics Generator – Finding an Optimal Plan. In the last step, a maximal compression ratio plan is selected upon dynamically computed statistics. In our system they must be generated for each metric and rolled time period. Pre-computing them and storing aside is not an optimal solution due to necessity of constant update and allocation of additional memory. Therefore all necessary estimations are calculated during this stage. Please note that if a plan contains a transformation algorithm then it must be applied before calculating statistics because it influences data.

Estimation results heavily depend on compression algorithms parameters. In [8] the choice of optimal parameters was straightforward, because used algorithms supported only compression of value to byte-aligned size (which reduced number of parameters) and did not allow exceptions in data (only one set of parameters was correct). However, in compression algorithms and compression plans which use PATCH mechanism, optimal parameter selection is more complex. Factors such as the number of generated exceptions and estimated exception compression size should be taken into account. For example, following data frame $\{1, 2, 3, 2, 32, 3, 3, 1, 64, 2, 1, 1\}$ could be compressed using PFL algorithm using 2 bits, 5 bits or 6 bits fixed-length, generating two exceptions (32, 64), one exception (64) or no exceptions. In this case, for each compression plan (selected in previous stages) a satisfactory set of parameters should be selected in order to correctly estimate compressed data size. This kind of computationally intensive task is ideal for parallel processing on a GPU device.

The following algorithms are used to calculate statistics.

- Bit histogram – used in size estimation of (P)FL and (P)FOR (includes estimation size of PATCH arrays with and without compression). Implemented with double buffering (registers and shared memory).
- Dictionary counter – used in size estimation of (P)DICT (includes estimation size of PATCH arrays with and without compression). As a side effect dictionary is generated for further usage if needed. Implemented with sort and reduction operations.
- Run length counter – used in RLE and PCONST. Implemented with reduction operation on key-value pairs.

All the above procedures were implemented using GPU parallel primitives mostly with CUDA Thrust library assuring the best performance. After statistics calculation step, the data is located in a GPU device memory and can be compressed without additional costs associated with the data transfer.

A complete plan evaluation must include base compression algorithm and dedicated helper algorithms sets. In case of all base algorithms, except for RLE, the helper compression algorithms appearing in the plan are already taken into account in the statistics. RLE requires to perform compression and then calculate statistics for the helper algorithms. For example, let us consider the following compression plan $[[\text{SCALE}, \text{DELTA}], [\text{PFL}], [\text{FL}, \text{FL}]]$ (notation – [transformation algorithms, base compression, helper methods]), first we apply transformation algorithms before estimating base algorithm compression size. Let us denote the data after applying the transformation algorithms by $(x_i)_{i \in I}$. For $1 \leq j \leq 32$ let $g(j) = \#\{i \in I : j \text{ bits are sufficient to write } x_i\}$. The size of the data after compression using remaining part of plan (i.e. $[[\text{PFL}], [\text{FL}, \text{FL}]]$) is then estimated by

$$E := \min_{1 \leq j \leq 32} \left(\sum_{l=1}^j g(l) \cdot \text{len}(g) + \sum_{l=j+1}^{32} g(l) (\lceil \log_2 \text{len}(g) \rceil + \text{last}(g) - j) \right),$$

where $\text{len}(g) = \sum_{l=1}^{32} g(l)$ and $\text{last}(g) = \max_{1 \leq l \leq 32} \{l : g(l) \neq 0\}$. First sum estimates base algorithm compression size and second estimates compression size of two exception arrays compressed using FL algorithm. If we change PFL to PFOR similar estimation is made but in first step $\min(x_i)_{i \in I}$ is subtracted from all values. PDICT works on dictionary counter array and uses it to build an optimal dictionary with exceptions (i.e. PDICT generate three output arrays and each may be compressed using FL, optimal dictionary with exception is such that minimizes estimated compression size after applying PDICT algorithm and using FL helper algorithm). Detailed description of other evaluation functions is beyond the size limitation of this paper and will be published separately.

3 Runtime Results

We compared effectiveness of dynamic compression planner and a single static plan within the same CF (Column Family – portion of data rolled in a database) by running the prototype system on samples from a set of network servers monitoring. The data included memory usage, the number of exceptions reported, services occupancy time or CPU load. Data covered a sample of 20 days of constant monitoring and contained about 91K data points in just a few time series (available at www.mat.umk.pl/~eror/gid2013). It was taken as a very short and limited sample from a telecommunication monitoring system which collects about 700.000 data points per day. Please note that in this case quality of the sample (its origin) is more important than its length.

We used the following equipment: *Nvidia*® *Tesla C2070 (CC 2.0)* with 2687 MB; 2 x Six-Core processor *AMD*® *Opteron*™ with 31 GB RAM, *Intel*® RAID Controller *RS2BL040* set in RAID 5, 4 drives *Seagate*® *Constellation ES ST2000NM0011* 2000 GB, Linux kernel 2.6.38–11 with the CUDA driver version 5.0.

3.1 Evaluation of Compression Planer

The evaluation was divided into two parts. The first measured efficiency of dynamic planner and was intended to prove the basic contribution of this work. The second checked efficiency of GPU based statistics evaluation when compared to CPU and proving contribution concerning time efficiency.

Figure 1 on the right shows compression ratio (original size / compressed size) using several static plans (one compression plan for the whole column family) and dynamic plan (dynamically chosen compression plan for different metrics, tags and time ranges). In case of timestamps, five static plans were generated using DELTA algorithm combined with five base compression methods (and helper compression algorithms if suitable). Similarly, for data values five plans were selected except SCALE was used instead of DELTA. We may observe, that for timestamp arrays, compression ratio of dynamic compression plan was equivalent to best static compression plan. This situation appeared because all time series were evenly sampled in this case. Therefore one static

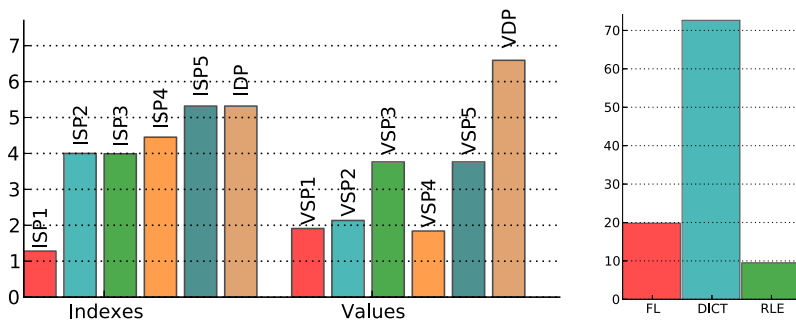


Fig. 1. Efficiency of the prototype dynamic compression system working on GPU. (left) Compression ratio for static (SP*) and dynamic (DP*) plans. I stands for index and V for values. (right) Statistics calculation speed-up including GPU memory transfer and using sample data with 8M values. (higher is better)

Table 2. Achieved bandwidth of pure compression methods (no IO)

Algorithm	DELTA	SCALE	(P)DICT	(P)FOR	(P)FL	RLE	PCONST
GB/s	28.875	41.134	6.924	9.124	9.375	5.005	2.147

plan for all metrics generated the same results as dynamic plan, selected for each time series separately. Note that in real systems, some measurements may be event-driven and thus dynamic plan could generate better results.

For data values, dynamic compression plan almost doubles compression ratio of best static compression plan which means that dynamic tuning was much better than selection of one static plan for the whole buffered column family. Obviously, this is heavily data dependant, but as a general rule dynamic compression plan will never generate a compression plan worse than the best static plan (as it always minimizes locally). Additionally hints system may be used to enforce static compression plan for cases when using dynamically generated compression plan does not produce satisfactory profits.

In Fig. 1 on the left GPU statistic generator is compared to similar CPU version (implemented as a single thread). A significant speed-up of factors from 10 to 70 was gained which guarantees no slowdown in a lightweight compression application.

4 Conclusions and Future Research

We successfully extended results from [8,12] by introducing three new implementations of patched compression algorithms on GPU (i.e. Patched DICT, Patched Const. and Patched Fixed Length). Furthermore we presented a dynamic compression planner adapted to time series compression in a NoSQL database. Our planner uses statistics calculated on the fly for the best plan selection. Resulting compression ratios and algorithms bandwidth combined with ultrafast decompression [8,12] on GPU are attractive solutions for databases.

Our future work will concentrate on query optimization in hybrid CPU/GPU environment, query execution on partially compressed data and extending dynamic compression planner by introducing additional costs factors (i.e. decompression execution time[8] or potential of query execution on compressed data).

References

1. Apache HBase (2013), <http://hbase.apache.org>
2. ParStream - website (2013), <https://www.parstream.com>
3. TempoDB – Hosted time series database service (2013), <https://tempo-db.com/>
4. Boncz, P.A., Zukowski, M., Nes, N.: Monetdb/x100: Hyper-pipelining query execution. In: CIDR, pp. 225–237 (2005)
5. 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. In: OSDI 2006: Seventh Symposium on Operating System Design and Implementation, Seattle, WA, pp. 205–218 (November 2006)
6. Cloudkick. 4 months with cassandra, a love story (March 2010), https://www.cloudkick.com/blog/2010/mar/02/4_months_with_cassandra/
7. Delbru, R., Campinas, S., Samp, K., Tummarello, G.: Adaptive frame of reference for compressing inverted lists. Technical report, DERI – Digital Enterprise Research Institute (December 2010)
8. Fang, W., He, B., Luo, Q.: Database compression on graphics processors. Proceedings of the VLDB Endowment 3(1-2), 670–680 (2010)
9. Fink, E., Gandhi, H.S.: Compression of time series by extracting major extrema. *J. Exp. Theor. Artif. Intell.* 23(2), 255–270 (2011)
10. Lees, M., Ellen, R., Steffens, M., Brodie, P., Mareels, I., Evans, R.: Information infrastructures for utilities management in the brewing industry. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM 2012 Workshops. LNCS, vol. 7567, pp. 73–77. Springer, Heidelberg (2012)
11. OpenTSDB. Whats opentsdb (2010-2012), <http://opentsdb.net/>
12. Przymus, P., Kaczmarek, K.: Improving efficiency of data intensive applications on GPU using lightweight compression. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM 2012 Workshops. LNCS, vol. 7567, pp. 3–12. Springer, Heidelberg (2012)
13. Przymus, P., Rykaczewski, K., Wiśniewski, R.: Application of wavelets and kernel methods to detection and extraction of behaviours of freshwater mussels. In: Kim, T.-h., Adeli, H., Slezak, D., Sandnes, F.E., Song, X., Chung, K.-i., Arnett, K.P. (eds.) FGIT 2011. LNCS, vol. 7105, pp. 43–54. Springer, Heidelberg (2011)
14. Yan, H., Ding, S., Suel, T.: Inverted index compression and query processing with optimized document ordering. In: Proc. of the 18th Intern. Conf. on World Wide Web, pp. 401–410. ACM (2009)
15. Zukowski, M., Heman, S., Nes, N., Boncz, P.: Super-scalar ram-cpu cache compression. In: ICDE 2006. Proc. of the 22nd Intern. Conf. on Data Engineering, p. 59. IEEE (2006)

Online Document Clustering Using GPUs*

Benjamin E. Teitler, Jagan Sankaranarayanan, Hanan Samet,
and Marco D. Adelfio

Department of Computer Science
University of Maryland
College Park, Maryland 20742 USA
{bteitler,jagan,hjs,marco}@cs.umd.edu

Abstract. An algorithm for performing online clustering on the GPU is proposed which makes heavy use of the atomic operations available on the GPU. The algorithm can cluster multiple documents in parallel in way that can saturate all the parallel threads on the GPU. The algorithm takes advantage of atomic operations available on the GPU in order to cluster multiple documents at the same time. The algorithm results in up to 3X speedup using a real time news document data set as well as on randomly generated data compared to a baseline algorithm on the GPU that clusters only one document at a time.

Keywords: Document Clustering, GPU, NewsStand, TwitterStand.

1 Introduction

Our work on indexing spatial and temporal data [1,4,5,6] and similarity searching [3,7,9] in the serial domain as well as in a distributed domain [10] and on GPUs [2] has led us to work on indexing textual representations of spatial data found in documents such as news articles [11] and tweets [8] to be accessed using a map query interface. A key piece of technology that makes all these systems work is an online clustering algorithm that takes news articles and noisy tweets as input streams and aggregates them into news topics. As news articles and Tweets enter our system as an input stream, we assign them to news clusters, which is a one-shot process, meaning that once an article is added to a cluster, it remains there forever. We will never revisit or recluster the news article, which is desirable because articles and Tweets are coming at a high throughput rate, and we need a fast and efficient clustering system that maintains good quality clustering output. In other words, our clustering algorithm is an online algorithm, and the additional constraints imposed on this problem add new complexity. Our clustering algorithm is different from traditional document clustering algorithms (such as the ones used by Google News) as we do not have access to the entire

* This work was supported in part by the National Science Foundation under Grants IIS-08-12377, IIS-09-48548, IIS-10-18475, and IIS-12-19023; and by Google and NVIDIA. J. Sankaranarayanan is currently at NEC Labs.

data set at the start of the algorithm. In particular, our online clustering algorithm is a leader-follower type algorithm [8,11], which means that our similarity function takes into account both content as well as the publishing time.

The focus of this paper is on developing clustering methods that are both online in nature as well as being able to take advantage of the parallelism and computational intensity afforded by Graphical Processing units in order to keep pace with the rate of arrival of these news articles. To cope with the high rate of our input stream and the need to process the input quickly in one shot, requires the mapping of the online clustering algorithm to the GPU in order to achieve a reasonable speed-up versus a CPU only implementation. Our clustering implementation uses the vector space model to represent documents, and makes use of the popular TF-IDF (term frequency inverse document frequency) method for computing term weights. We use the Euclidean dot product as our similarity metric between document vectors and cluster vectors. Online clustering is a challenging problem for the GPU because it is bandwidth intensive as opposed to being only computationally intensive. Fast document clustering requires maintaining an index on the clusters associated with every term in the document corpus. This allows for fast pruning of clusters that have no terms in common with a given document. This index is large and must be stored in GPU memory. The index is highly dynamic, and new parallel algorithms must be developed to update the index in an efficient manner. Another challenge that we face is how to evenly assign computations to each thread, as the work associated with each document to be clustered is extremely variable. Finally, other challenges emerge when the entire corpus cannot fit into GPU main memory.

Online document clustering takes as its input a list of document vectors, ordered by time. A document vector consists of a list of K terms and their associated weights. The generation of terms and their weights from the document text may vary, but the TF-IDF (term frequency-inverse document frequency) method is popular for clustering applications [12]. The assumption is that the resulting document vector is a good overall representation of the original document. We note that the dimensionality of the document vectors is very high (potentially infinite), since a document could potentially contain any word (term). We also note that the vectors are sparse in the sense that most term weights have a zero value. We assume that any term not explicitly present in a particular document vector has a weight of zero. Document vectors are normalized. In addition, clusters are represented as a list of weighted terms. At any given time, a cluster's term vector is equal to the average of all the document vectors contained by the cluster. Cluster term vectors are truncated to the top K terms (those containing the highest term weights) and then are normalized. The objective of the algorithm is to partition the set of document vectors into a set of clusters, each cluster containing only those documents, which are similar to each other with respect to some metric. For this paper, we consider the Euclidean dot product as the similarity metric, as it has been shown to provide good results with the TF-IDF metric [12]. The similarity between a cluster and a document is defined as the dot product between their term vectors.

The rest of the paper is organized as follows: Section 2 presents a sequential algorithm for online clustering. Section 3 describes a PRAM algorithm for parallel online clustering one document at a time using a CRCW programming model. Section 4 presents a practical implementation of a parallel online clustering algorithm, which clusters multiple documents at a time suitable for the CUDA parallel computing architecture [13]. Experimental results are presented in Section 5. Concluding remarks are provided in Section 6.

2 Sequential Clustering on the GPU

We first present a simple algorithm to cluster documents on the GPU one document at a time. This algorithm also serves as a baseline for our main algorithm that will be presented later that can cluster multiple documents at the same time. The basic sequential online clustering algorithm takes as input a list of n document vectors, as well as a clustering threshold T ranging between 0 and 1. Below is a high level overview of the algorithm.

```

For each document  $D$  (ranging from 0 to  $n - 1$ )
  Choose the cluster  $C$  most similar to  $D$ 
  if  $similarity(C, D) > T$  then
    | Add document  $D$  to cluster  $C$ 
    | Recompute  $C$ 's term vector
  else
    | Create a new cluster consisting of only the document  $D$ 
  end

```

Algorithm 1. Sequential Clusterer 1 on the GPU

In the worst case, Algorithm 1 takes $O(n^2)$ dot products to cluster n documents as each document could end up forming its own cluster. However, the sparseness of document vectors means that very few number of distance computations are needed per document [2,14]. Most document vectors have very few terms in common with other document vectors. Therefore, for each term in document vector D , we will have a limited number of clusters whose term vector contains a non-zero weight for that term. By keeping a list of clusters for each unique term seen by the clustering algorithm so far, we can reduce the number of dot products needed per document to only those dot products that will be non-zero. Let $D[t]$ represent the weight of term t for document D (the weight associated with t in D 's term vector). Similarly, let $C[t]$ represent the weight of term t for cluster C . We can avoid unnecessary work within dot products by keeping the term weight in each term list with its corresponding cluster. For instance, the term list for term t is: $TermList[t] = (C_1, C_1[t]), (C_2, C_2[t]), \dots (C_p, C_p[t])$.

This indicates that cluster C_i contains a non-zero weight for term t . Adding the weight information to the term list allows us to compute only the non-zero partial dot products between documents and clusters efficiently, since we have no need to look up t 's weight in C_i 's term vector. We describe a sequential algorithm on the GPU in Algorithm 2 which makes use of the TermList data structure. Note that we use D both to refer to the document and its term vector.

```

TermList  $\leftarrow$  Set of empty lists
for each document  $D$  (ranging from 0 to  $n - 1$ ) do
  Candidates  $\leftarrow$  Empty Set
  Results  $\leftarrow$  Array of size  $D$ , initialized to all 0
  for each term  $t$  in  $D$ 's term vector do
    for each  $(C_i, C_i[t])$  in TermList[  $t$  ] do
      Results[  $C_i$  ] = Results[  $C_i$  ] +  $C_i[t] * D[t]$ 
      if Candidates does not contain  $C_i$  then
        | Add  $C_i$  to Candidates
      end
    end
  end
  Choose the cluster  $C$  in Candidates with the max(Results[ $C$  ])
  if similarity( $C, D$ ) >  $T$  then
    for each term  $t$  in  $C$ 's term vector do
      | Remove  $C$ 's entry  $(C, C[t])$  from TermList[ $t$ ]
    end
    Add document  $D$  to cluster  $C$  and recompute  $C$ 's term vector
  else
    | Create a new cluster  $C$  consisting of only the document  $D$ 
  end
end

```

Algorithm 2. Sequential Clusterer 2 on the GPU

We calculate the approximate running time cost of Algorithm 2 as follows. Recall that K is the number of terms kept in each of the document and cluster term vectors. Let L represent the average number of clusters that contain any given term t at any specific time in the clustering algorithm. This indicates that to cluster any given document D , we have roughly $K * L$ partial dot product computations. We also have at most $K * L$ insertions into the Candidates set, each taking $O(1)$ time using a hash set implementation. We have at most K deletion and K insertions from lists of size L , in order to update the TermList data structure. Assuming an array data structure for each TermList[t], we have $O(1)$ insertion and $O(L)$ deletion for each term, and the run-time of the algorithm is given by $O(n * K * L)$. We note that although L is highly dependent on the dataset, it is expected to be far less than n . The memory required for Algorithm 2 is $O(m * K)$, where m is the total number of clusters at the end of the algorithm.

3 Parallel Clustering of a Single Document

We first consider the case of parallelizing the work associated with clustering a single document, while still clustering each of the n documents sequentially. Later we will discuss the case of processing multiple documents in parallel, and its effects on the clustering output.

Our goal is to parallelize as much of sequential clusterer's document loop as possible. We first note that the dot product operations are highly parallelizable. All the partial dot product operations for a given document can be done in

```

TermList  $\leftarrow$  Set of empty lists
for each document  $D$  (ranging from 0 to  $n - 1$ ) do
  Partials  $\leftarrow$  Array initialized to all 0
  Let  $t_1, t_2, \dots, t_K$  be the terms in  $D$ 's term vector
   $S \leftarrow (t_1 \times \text{TermList}[t_1]) \cup \dots \cup (t_K \times \text{TermList}[t_K])$ 
  for each  $(t_i, C_i, C[t_i])$  in  $S$  parallel do
    | Partials[ThreadID] =  $(C_i, D[t_i] * C[t_i])$ 
  end
  Run parallel sort on Partials, sorting by  $C_i$ 
  Run parallel summation on Partials (adding similar  $C_i$ )
  Run parallel max on Partials to produce best candidate cluster  $C$ 
  if  $\text{similarity}(C, D) > T$  then
    | for each term  $t$  in  $C$ 's term vector do
      | | Remove  $C$ 's entry  $(C, C[t])$  from TermList[ $t$ ]
    | end
    | Add document  $D$  to cluster  $C$  and recompute  $C$ 's term vector
  else
    | Create a new cluster  $C$  consisting of only the document  $D$ 
  end
  Add document terms in  $C$  to TermList
end

```

Algorithm 3. Parallel Clusterer Algorithm 1 on the GPU

parallel. We can then run a parallel sorting operation with the cluster as the sorting key. Finally, we run a parallel summation operation to gather the completed dot products for each cluster, followed by a parallel maximum operation to choose the cluster with best similarity to D . After the best cluster C has been chosen, we must update our TermList data structure to reflect the changes to C 's term vector. We first delete the old TermList entries of C by assigning a different processor to look at each entry of TermList[t], for every term t in C . Processors that find their entry $(C_i, C_i[t])$ swap in the last value of the TermList[t] to compact that list (assuming an array implementation). Inserting the new $(C_i, C_i[t])$ values can be done trivially by assigning K processors to add the new $(C_i, C_i[t])$ to the end of their respective lists.

We now present a high level parallel algorithm for clustering in Algorithm 3. We introduce the parallel keyword to indicate that the contents of a loop are performed in parallel. We also introduce a value ThreadID which is available to each thread within a parallel loop. For h threads, the values of ThreadID range between 0 and $h - 1$ inclusively. Assume that each parallel thread is assigned a unique ThreadID value. We use a PRAM architecture using the CRCW (Concurrent read-concurrent write) model [15] to analyze the run-time of parallel algorithms even though the GPU has a less restrictive computation model.

We can estimate the running time of Algorithm 3 as follows. For each document, we can compute the partial dot products in $O(1)$ time by using $K * L$ processors (we ignore the complication here of assigning ThreadIDs to processors). Parallel sort is known to be logarithmic [14], and so this takes $O(\log(K * L))$ time. Parallel summation of Partials can be done in $O(\log(K * L))$ time, and

the parallel max operation also takes $O(\log(K * L))$ time. Finally, the TermList maintenance operations take $O(1)$ time each. The running time for the algorithm is $O(n * \log(K * L))$. The memory require for Algorithm 3 is again $O(m * K)$.

We now discuss the process of assigning ThreadIDs to processors for Algorithm 3's partial dot product computation. Recall that we have specified L as the average size of TermList[t] for any given term t . This is useful for analyzing running time, but the sizes of TermList[t] will vary greatly when clustering a specific document, which complicates the ThreadID assignment. Our goal is to decide on a specific $(t_i, C_i, C[t_i])$ to associate with every ThreadID. This requires deciding on one specific element of each TermList[t] for each ThreadID. Let TermList[t][j] refer to the j -th element of term list for term t . Let size (TermList[t]) represent the number of elements currently in the term list for t .

```

Let  $t_1, t_2 \dots t_K$  be the terms in  $D$ 's term vector
TermSizes  $\leftarrow$  size(TermList[ $t_1$ ])  $\dots$  size(TermList[ $t_K$ ])
PrefixSums  $\leftarrow$  the prefix sums of TermSizes
Binary search on PrefixSums to find the smallest  $u$  s.t., ThreadID <
PrefixSums[ $u$ ]
 $C =$  TermList[ $t_u$ ][ PrefixSums[ $u$ ] - ThreadID - 1 ]
Partials[ThreadID] = (C, D[ $t_u$ ] * C[ $t_u$ ])
    
```

Algorithm 4. Thread assignment

We note that PrefixSums[$i - 1$] indicates how many threads should be assigned to term lists 1 up to $i - 1$. This means that the term u assigned to a given ThreadID is simply the first u such that ThreadID < PrefixSums[u]. The value PrefixSums[u] - ThreadID - 1 gives us the index into TermList[u] in which we are interested. Each binary search using Algorithm 4 takes $O(\log(K))$ time. Binary searches over global memory arrays can be inefficient. The performance can be improved by using an additional temporary array and another PrefixSum, which has much better locality and therefore processes data faster. While the complexity of PrefixSum is $O(\log n)$, it does not change the overall running time of Algorithm 3, since it is dominated by the cost of sorting. Finally, we note that the parallel deletion that occurs in Algorithm 3 requires an identical ThreadID configuration as the partial dot products. Each deletion thread will receive a unique ThreadID, and must decide which TermList entry to examine. We can use Algorithm 4 where $t_1, t_2 \dots t_K$ are the terms in C 's term vector, instead of D 's. Again, the overall running time is unchanged.

4 Clustering Multiple Documents in Parallel

In this section we examine an algorithm for clustering multiple documents in parallel. Assume that we wish to cluster Q documents in parallel. We define the multiple document clustering algorithm below as Algorithm 5.

The main difference between the multiple document and single document algorithms is that we assign the best clusters to Q documents before updating the cluster term vectors and the index. This can lead to poor clustering in some cases, since, for example, document D_i is never compared against the effects

```

while we have more documents to cluster do
    Choose the next  $Q$  documents  $D_1, D_2, \dots, D_Q$ 
    Choose clusters  $C_1, C_2, \dots, C_Q$  such that  $C_i$  is the most similar cluster to  $D_i$ 
    for  $i = 1$  to  $Q$  do
        if  $\text{similarity}(C_i, D_i) > T$  then
            Add document  $D_i$  to cluster  $C_i$  and recompute  $C_i$ 's term vector
        end
    end
end

```

Algorithm 5. Multiple Document Clusterer 1

of $D_1, D_2 \dots D_{i-1}$. Merging similar clusters at varying points in the algorithm can possibly mitigate this effect. We assume that the effects of this problem are minimal as long as Q is much less than n . We wish to extend Algorithm 3 to cluster Q documents in parallel. We first note that computing the partial dot products for Q documents can be done using Q parallel instances of the single document version of the dot product computation. However, assigning ThreadIDs for multiple documents now requires reasoning about to which document a thread belongs. This results in a binary search of a prefix sums array of size $K * Q$ for each thread to assign work. Assume that t_{ij} refers to the j -th term of document D_i (the j -th term of the i -th document that we are clustering in parallel). Note that each entry contained in `Partials` now contains an extra element, which indicates the document to which the partial dot product belongs. Algorithm 6 guarantees however that similar D_q values will be contiguous within `Partials`. This means that we can sort Q separate sub-lists in parallel (each of size roughly $K * L$).

```

TermSizes ← size(TermList[t11]) ··· size(TermList[t1K]),
              size(TermList[tQ1]) ··· size(TermList[tQK])
PrefixSums ← prefix sums of TermSizes
Binary search to identify smallest  $u$  s.t. PrefixSums ← prefix sums of TermSizes
Binary search to identify smallest  $u$  s.t. ThreadID < PrefixSums[ $u$ ]
 $q \leftarrow u / K$ 
 $r \leftarrow u \% K$ 
 $C = \text{TermList}[tqr][\text{PrefixSums}[u] - \text{ThreadID} - 1]$ 
Partials[ThreadID] = ( $D_q, C, D[tqr] * C[tqr]$ )

```

Algorithm 6. ThreadID Assignment 3

Once, we haven chosen the appropriate clusters $C_1, C_2 \dots C_Q$, we must update the Term-List data structure to reflect the changes of the Q cluster term vectors. We cannot simply perform these operations in parallel for all Q documents as in the single document case, since different clusters may have terms in common. This means that they will update the same `TermList[t]` and interfere with each other. To deal with this issue, we use the atomic addition operator available in CUDA, while acknowledging that their frequent use can result in performance degradation. Parallel insertion of a term t , and an element to insert x is given by: `size = atomicAdd(TermList[t].size, 1)` followed by `TermList[t][size] = x`. Each

thread that attempts to insert into a given $\text{TermList}[t]$ will receive a unique slot to receive its element. After all insertions have been completed, the new size for $\text{TermList}[t]$ is the new size of the list. The parallel deletion algorithm is a little more involved and is given below as Algorithm 7.

```

TermList[t].deleteNumber = 0
TermList[t].deletePriority = 0
TermList[t].newSize = TermList[t].size
atomicAdd(TermList[t].deleteNumber, 1 )
atomicAdd(TermList[t].deletePriority, 1 )
atomicAdd(TermList[t].newSize, -1 )
for  $i = 0$  upto  $\text{TermList}[t].\text{size} - 1$  parallel do
  TermList[t][i].deleted = FALSE
  if  $\text{TermList}[t][i] == x$  then
    TermList[t][i].deleted = TRUE
    if  $i \geq \text{TermList}[t].\text{size} - \text{TermList}[t].\text{deleteNumber}$  then
      | Return
    end
    priority = atomicAdd(TermList[t][i].deletePriority, -1 )
    numSkip = TermList[t].deleteNumber - TermList[t].priority
     $j =$  elements from end of  $\text{TermList}[t]$  s.t.,  $\text{TermList}[t].\text{deleted}$  is FALSE
    TermList[t][i] = TermList[t][j]
  end
  TermList[t].size = TermList[t].newSize
end

```

Algorithm 7. AtomicDeletion(t, x)

The basic idea behind Algorithm 7 is to assign a priority to each thread that finds an element to delete. Based on this priority, the thread picks the correct element near the end of the list to move into the hole created by the deleted element. This algorithm assumes each parallel call to AtomicDeletion has a unique (t, x) (no call has both the same t and x as another call). This is a valid assumption, since we can prune C_i values that are duplicates prior to running the AtomicDeletion, as the result of including them is the same as that when we don't include them.

Due to space limitations, the details of the Multiple Document Clusterer 2 are provided in [16]. The running time of this algorithm is given by $O((n/Q) * \max(\log(K * L), Q))$, while the memory requirement is $O(\max(m * K), K * L * Q)$.

5 Experimental Results

First, we evaluated the performance of Algorithm 2 on a real world dataset, which consists of news documents from a span of 90 days taken from a wide variety of news sources. The result is shown in Figure 1a. The documents are ordered by the time of publication. Each news document contains 20 terms in its term vector ($K = 20$). Our implementation is written in C++ and compiled using g++ (GCC) version 4.1.2 with the $-O3$ optimization flag. We tested our implementation on a GeForce GTX 280, which has 240 cores and 1 GB of global

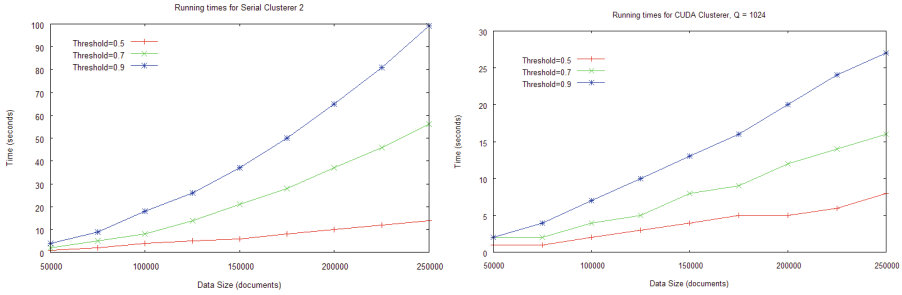


Fig. 1. Running time of a) Sequential Algorithm 2 b) Multiple Clusterer 2 for different thresholds and data sizes

memory. The CPU was an AMD 3GHz processor with 4 cores. It can be seen that the algorithm takes about 50 seconds to cluster 250k documents.

Next, we performed clustering of more than one document at a time using the Multiple Document Clusterer 2 algorithm. Figure 1b is the result for $Q = 1024$ (1024 documents done in parallel). It can be seen that the algorithm takes only 15 seconds to cluster 250k documents. In contrast the Sequential Clusterer on the GPU takes only 50 seconds denoting a 3X speed up by performing clustering in parallel. Furthermore, we note that the best speedup is achieved using the highest clustering threshold. This is expected as a higher clustering threshold means there will be more clusters, and therefore more cluster candidates per document (more non-zero partial dot products).

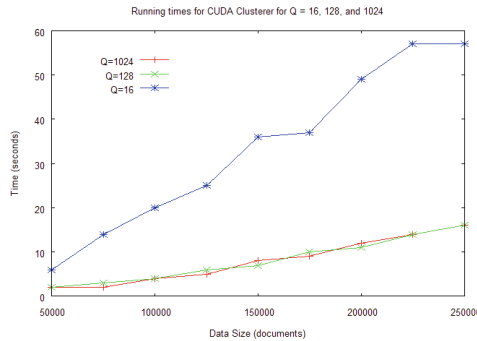


Fig. 2. Running time of Multiple Clusterer 2 for different values of Q

Finally, we compare the running times of the CUDA Clusterer for three different values of Q (16, 128, and 1024) using a threshold of 0.7. We observe from Figure 2 that there is a significant performance improvement in increasing the value of Q from 16 to 128. However, increasing the value of Q more does not result in significant reduction of running time. This indicates that the GPU's threads have saturated when the number of documents is above 16. Note however that setting a large value of Q does not seem to have a detrimental effect on the running time of the algorithm.

6 Concluding Remarks

In this paper we have described a parallel algorithm for online document clustering. We have shown that 3X speedups can be achieved when clustering multiple documents at the same time instead of one at a time. Future work will focus on incorporating the algorithm into our NewsStand and TwitterStand production systems and developing a variant of the algorithm that makes limited use of atomic operations.

References

1. Hjaltason, G.R., Samet, H.: Speeding up construction of PMR quadtree-based spatial indexes. *VLDB Journal* 11(2), 109–137 (2002)
2. Lieberman, M.D., Sankaranarayanan, J., Samet, H.: A fast similarity join algorithm using graphics processing units. In: *IEEE ICDE*, pp. 1111–1120 (April 2008)
3. Samet, H.: K-nearest neighbor finding using MaxNearestDist. *IEEE TPAMI* 30(2), 243–252 (2008)
4. Samet, H., Alborzi, H., Brabec, F., Esperança, C., Hjaltason, G.R., Morgan, F., Tanin, E.: Use of the SAND spatial browser for digital government applications. *CACM* 46(1), 63–66 (2003)
5. Samet, H., Rosenfeld, A., Shaffer, C.A., Webber, R.E.: A geographic information system using quadtrees. *Pattern Recognition* 17(6), 647–656 (1984)
6. Samet, H., Tamminen, M.: Bintrees, CSG trees, and time. *Computer Graphics* 19(3), 121–130 (1985); also in *SIGGRAPH* 1985
7. Sankaranarayanan, J., Alborzi, H., Samet, H.: Efficient query processing on spatial networks. In: *ACM GIS*, pp. 200–209 (November 2005)
8. Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D., Sperling, J.: Twitterstand: News in tweets. In: *ACM GIS*, pp. 42–51 (November 2009)
9. Sankaranarayanan, J., Samet, H., Varshney, A.: A fast all nearest neighbor algorithm for applications involving large point-clouds. *Computers & Graphics* 31(2), 157–174 (2007)
10. Tanin, E., Harwood, A., Samet, H.: A distributed quadtree index for peer-to-peer settings. In: *IEEE ICDE*, pp. 254–255 (April 2005)
11. Teitler, B.E., Lieberman, M.D., Panozzo, D., Sankaranarayanan, J., Samet, H., Sperling, J.: NewsStand: A new view on news. In: *ACM GIS*, pp. 144–153 (November 2008)
12. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24(5), 513–523 (1988)
13. Sanders, J., Kandrot, E.: *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional, Reading (2011)
14. Weber, R., Schek, H.-J., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high dimensional spaces. In: *VLDB*, pp. 194–205 (August 1998)
15. Vishkin, U.: *Thinking in parallel: Some basic data-parallel algorithms and techniques*, College Park, MD (2007)
16. Teitler, B.E., Sankaranarayanan, J., Samet, H.: Online document clustering using the GPU. *CS-TR 4970*, UMD, College Park, MD (August 2010)

Part IV

The Second International Workshop on Ontologies Meet Advanced Information Systems

Using the Semantics of Texts for Information Retrieval: A Concept- and Domain Relation-Based Approach

Davide Buscaldi², Marie-Noëlle Bessagnet¹, Albert Royer¹,
and Christian Sallaberry¹

¹ LIUPPA, Université de Pau et des des Pays de l'Adour, F-64000 Pau
{marie-noelle.bessagnet,albert.royer,christian.sallaberry}@univ-pau.fr

² LIPN, Université Paris XIII, F-93430 Villetaneuse
davide.buscaldi@lipn.univ-paris13.fr

Abstract. Our hypothesis is that assessing the relevance of a document with respect to a query is equivalent to assessing the conceptual similarity between the terms of the query and those of the document. In this article, we therefore propose a method of calculating conceptual similarity. Our information retrieval strategy is based on exploring an ontology and domain relations between concepts marked by verbal forms. Our approach overall is implemented by a prototype and the results obtained are evaluated. We thus show that a semantic IR system based on concepts improves recall with respect to a classic IR system and that a semantic IR system based on concepts and domain relations improves precision with respect to IR based on concepts alone.

Keywords: information retrieval, ontology, similarity measure.

1 Introduction

In the last decade, the amount of digital information in the world has been continuously growing, boosted by technological advances. More and more data are published on the web; for instance, the number of articles in the English Wikipedia is now about 4 million pages, compared to the 19,700 it contained ten years ago¹. In order to deal with this explosion of data, search engine technology has experienced some important enhancements. However, these enhancements are still limited by the use of keywords, in contrast with the idea of “conceptual” search, where the basic item indexed and searched is a concept (representing the meaning of a word or a phrase). This “conceptual” search paradigm is often referred to as *Semantic Information Retrieval* (SIR).

The use of semantics to enhance IR techniques, by outstripping search models based on keywords, is an open research topic, which is drawing the attention of a large number of researchers from different fields: Information Retrieval, Knowledge Representation and Management and the Semantic Web (SW). Since the

¹ http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

research involving this topic has been carried out from such different perspectives, the wording “Semantic search” has been used in a variety of different tasks. Semantics can be specified explicitly, using a formal representation of knowledge like an *ontology* - in this case the expression “ontology-based search” is also used - or implicitly, deriving concepts from the distribution of words in text collections (for instance, using Latent Semantic Analysis [5] or Explicit Semantic Analysis [8]). If an ontology is used, it could be used in different ways, usually depending on whether the developers’ vision is oriented towards the SW or the SIR perspective. For instance, in the SW perspective, an ontology is used as a knowledge database that can be queried, transforming a natural language request in a SPARQL query, finding the result in the database itself. This can be also viewed as a form of Ontology-based Question Answering (QA), where the answer to a user request is found directly in the ontology (see the Yago-Naga project² or Broccoli³). On the other hand, according to the SIR perspective, an ontology is used as a source of knowledge that can be exploited to expand concepts (in the original query and/or in documents) with semantically related concepts, navigating the relations in the ontology. SIR systems can be also categorised into systems that target the web or a static textual collection. In this work we will adhere to the SIR perspective rather than the SW one.

In this article, we set out our SIR system and an evaluation of it. In Section 2 we present the related work and discuss our contributions. In Section 3 we motivate our approach to a semantic information search. In Section 4 we describe the main features of our proposal. In Section 5 we summarize the experiments and evaluation performed. Lastly, in Section 6 we draw our conclusions and propose some prospects.

2 Related Work and Our Contributions

A crucial step in all semantic IR tasks consists in the *annotation* of concepts in documents. Concepts can be identified automatically, using supervised or unsupervised methods, or manually, where an expert specifies rules and/or keywords that indicate the presence of a concept in a fragment of text. Ontologies can also be used to back annotation tools like Phenote⁴ or Brat⁵. Once the concepts have been identified in text, they can be used for the following tasks:

- Calculate the similarity between texts, taking into account semantic similarity measures based on the hierarchical or domain relations in the ontology;
- Create an expanded text index, that is, an index where the implicit information that can be derived from the concept is made explicit to enhance the search process.

² <http://www.mpi-inf.mpg.de/yago-naga/>

³ <http://broccoli.informatik.uni-freiburg.de>

⁴ <http://www.phenote.org/>

⁵ <http://brat.nlplab.org/>

We would like to underline the importance of the quality of the annotation in SIR: in fact, during the long history of concept-based retrieval research, it has been found that concept-based retrieval cannot really outperform term-based retrieval for the simple reason that a perfect concept-based representation is hard to construct automatically, especially in the case of open domain IR. However, some results like those obtained by the system proposed by [16] proved that domain-specific approaches can improve the results of classical IR systems, since it is easier to build a conceptual representation of a narrow domain. This is our case.

The automatic extraction of concept is used in [6], according to a model in which the association between concepts and keywords is established using unsupervised training on Wikipedia. In the work of [7], instead, Part-of-Speech (POS) tagging is used to tag words with their POS category; thanks to empirically identified POS patterns, the keywords representing potential concepts are searched in the ontology in order to find the matching concepts. The KIM system [11] works instead on Named Entities (NEs) only, such as person, location, date, organisation, etc., using NE recognition techniques and a custom knowledge base.

Recent works [3,10,13] have proposed the association of a terminological and/or linguistic part with the ontologies with a view to establishing a clear distinction between the terminological component and the conceptual component. [13] defines the notion of Ontological-Terminological Resource(OTR), in particular. These resources are usually developed by domain experts and can be used effectively for the annotation process since they identify with a high level of precision the keywords or keyphrases that fit to the concepts in the ontology.

Independently from the technique used to annotate concepts, most SIR systems use them as a “bag of concepts”, with ranking functions usually based on keyword-based approaches such as the vector space [7,2], or the probabilistic model [6]. The taxonomical structure of ontologies is also exploited in most systems, using some conceptual similarity measure to compare concepts. This allows to calculate a score even if the document does not contain the same concept of the request. We can name the conceptual similarity formulae [15,4], which are distance-based (that is, they assign a score only on the basis of paths); the approaches using the informational content of the concepts are based on corpus-collected statistics: among them, we may cite the works of [12] and [9]. The classic hierarchical relations, *part-of*, *is-a*, are not sufficient to express the semantics contained in documents and queries. It is therefore necessary to model semantic relations and to find ways of evaluating their semantic similarity.

Our work presents two main contributions: (1) an automatic annotation process carried out using an OTR; and (2) a ranking algorithm which takes into account also relationships (including domain relations) and not only concepts. Our IR approach has two steps: a first one, as other systems, computes documents-query similarity by taking into account concepts; the second step improves the

previous ranking with new scores produced by the domain relations existing both in documents and query.

3 Motivation

Suppose that we have, as a corpus, two texts (Figure 1), an ontology of botany and two systems by means of which we can annotate concepts with the first, and concepts and domain relations with the second.

Text1 : “Dans vos jardins, les glaïeuls s’épanouissent en juillet”
 (In your gardens, the gladioli flower in July)
 Text2 : “Arrosez copieusement vos glaïeuls dès le 1er juillet”
 (Water your gladioli liberally from 1st July)

Fig. 1. Example of corpus

In our ontology, we have the concepts of *Plant* and *Gladiolus*. We have the concept of *Period*, which is the group of the seasons, which themselves are made up of months. Thus *July* is a month in the season of *Summer*.

With a concept-based IR system, the concepts of *Plant* “*gladioli (glaïeuls)*” and *Period* “*July (juillet)*” are annotated in Text1, and the concepts of *Plant* “*gladioli (glaïeuls)*” and *Period* “*July (juillet)*” are annotated in Text2.

In the user query “what plants bloom in summer?” , the concepts *Plant* “*plant(plante)*” and *Period* “*summer(été)*” are annotated.

With an IR system based on concepts and relations, the domain relation *FlowersIn* will be detected between *Plant* and *Period* in Text1 “*flower(s’épanouissent)*” and in the query “*bloom(fleurissent)*”.

In the matching and IR phase,

- a classic IR system based on keywords searches for the three keywords “plant”, “bloom”, “summer” and will return no result.
- an IR system using concepts annotates the concepts *Plant* “plant” and *Period* “summer” and returns the two documents Text1 and Text2. Thus a system of this kind provides improved recall in comparison with a keyword IR.
- an IR system utilizing concepts and domain relations annotates the concepts *Plant* “plant” and *Period* “summer” as well as the domain relation *FlowersIn* “bloom” between *Plant* and *Period*. This system returns Text1 only. Thus such a system provides improved precision when compared with a system utilizing concepts alone.

These are the basic issues that we detail in the following sections.

4 IR Based on Concepts and Domain Relations

The SIR process we propose is composed of four main stages:

1. The indexing stage organizes the description of the collection of documents in the form of a semantic representation : concepts and domain relations are indexed. To identify concepts and domain relationships, the complete terminology is projected onto the text to be annotated (§4.2) during a back-office process.
2. Querying a documentary resource requires that the query should be represented in a compatible form. Concepts and domain relations of the query are annotated during this stage. The collection of documents and the user query are annotated by the same process (§4.2).
3. The query-document matching stage enables selection of a list of documents by examining the similarity of the concepts (formula 1, §4.3). This matching, based on similarity of concepts, calculates the score of relevant documents.
4. Then, the result list of documents is reordered according to the boost given by a second stage of matching that takes into account similarity of domain relations. The relations described in the index are compared with those annotated in the query to calculate the boost according to formula 2, §4.3.

4.1 Notions of Ontology, OTR, Domain Relation and Corpus

We adopt the following notations to formalize the process of annotating concepts and relations, on the one hand, and the matching process that utilizes annotations, on the other hand.

c	for a concept in the ontology,
d	for a document in the corpus,
f	for a field (title, section, paragraph) in a document,
r	for a domain relation in the ontology,
t	for a term in a document.

The concepts in our **ontology** are the classes of a specific domain; for example, in the botanical domain, the concept *gladiolus* is a class of which the plant *Colville's gladiolus* is a sub-class. In addition to the classic hierarchical relations such as *part-of* or *is-a*, domain relations are modeled; thus, we may note that *Colville's gladiolus* is *planted* in *October-November*.

For this reason, we define an ontology O by the set C of the concepts of the domain and by the set R of relations between concepts, and we write $O = (C, R)$. It should be noted that $R = \{r_\nu\}$ with $r_\nu = (\delta, \nu, \rho)$ where the relation r_ν named ν has as its class domain δ and as its class range ρ .

We assume that in an **OTR**, each concept is associated with a list of terms that *denote* the concept. We note T as the set of terms that can denote a concept.

With regard to **domain relations**, several predicates are required:

$has_label_r(r, t)$ for $r \in R$ and $t \in T$,

indicates that r bears as a label the term t ;

$relation(c_\delta, t, c_\rho)$ for $c_\delta \in C, t \in T$ and $c_\rho \in C$

indicates a relation revealed by t between c_δ and c_ρ ;

The **corpus** is seen as a set of documents D . Each document is made up of a number of fields. The set of n fields of a document $d \in D$ will be noted $F_d = \{f_0, \dots, f_n\}$. Moreover, each document has a *core* concept c_γ that corresponds to the class characterizing the main subject (assuming that the document is written in encyclopedic style, the core concept is extracted from the *title* field), for example, the class *Gladiolus* (Glaïeul).

4.2 Annotation of Concepts and Domain Relations

The note $T_{f,d}$ is given to the set of terms in the field f of the document d . **Annotation of concepts** is performed by fields. A field f contains a concept c if and only if a term t denoting a concept c' exists and if the concept c' is a descendent of c or if $c = c'$.

With regard to the **annotation of domain relations**, a field f contains a relation r in two cases:

- either if three terms of the field denote the relation and the concepts of the domain c_δ and the range c_ρ fitting this relation,
- or if two terms of the field denote the relation and the concept of the range c_ρ corresponding to the latter; the concept of the domain of the relation being the core concept c_γ .

This annotation process corresponds to the stages 1 and 2.

4.3 Matching of Concepts and Domain Relations

Matching based on similarity of concepts is described by formula 1. Given Q the set of concepts in a query and D the set of concepts in a document, and given $F(c)$ the function that gives the coefficient of dominance of a concept c (the coefficients are given by the user and saved in a configuration file), the weight for a document is calculated as follows:

$$w(Q, D) = \frac{\sum_{c_1 \in Q} (F(c_1) \cdot \max_{c_2 \in D} s(c_1, c_2))}{\sum_{c_1 \in Q} F(c_1)} \quad (1)$$

where $s(c_1, c_2)$ is the similarity measure.

This formula corresponds to a commonly used matching approach based on concepts which is detailed in [4].

Matching based on similarity of concepts and domain relations is described by formula 2. We take domain relations into account to extend this initial approach based on matching concepts.

Let R_Q be the set of relations r_1, \dots, r_k found in a query and denoting domain relations between concepts of the set Q . Each relation is a triplet $r = (c_\delta, \nu, c_\rho)$ where c_δ is the concept relating to the domain, ν the name of the relation and c_ρ

the concept relating to the range. We define $d(r) = c_\delta$, $n(r) = \nu$ and $e(r) = c_\rho$. Two relations r_1, r_2 are comparable only if $n(r_1) = n(r_2)$. We define the set R_D as the set of relations found in the document D . The weight of a document is calculated as $w(Q, D) + b(R_Q, R_D)$ where :

$$b(R_Q, R_D) =$$

$$\frac{\sum_{\substack{r_1 \in R_Q, r_2 \in R_D \\ \text{et } n(r_1) = n(r_2)}} (F(d(r_1)).s(d(r_1), d(r_2)) + F(e(r_1)).s(e(r_1), e(r_2)))}{\sum_{\substack{r_1 \in R_Q, r_2 \in R_D \\ \text{et } n(r_1) = n(r_2)}} (F(d(r_1)) + F(e(r_1)))} \quad (2)$$

We describe $b(R_Q, R_D)$ as a *boost* (cf. formula 2) which augments the weight of a document that contains all or some of the relations detected in R_Q , and consequently repositions such documents at the head of the resulting list. Here, for each pair of relations $r_1 \in R_Q, r_2 \in R_D$ of the same name, we calculate the similarity of the concepts relating to the domains and ranges of these relations respectively. The sum of these similarity measures, normalized by the corresponding dominance coefficients, determines the *boost*.

These similarity measures are implemented in our *ThemaStream* IR system: some experiments are described in the next section.

5 Experiments

We implemented this approach and then experimented it in the framework of the MOANO⁶ ANR project. In this project, our contribution concerns the indexing and retrieval of information based on semantic resources. The OTR resource built by our colleagues of the MELODI team (IRIT laboratory) depicts a domain specific ontology[1]. To this end, we designed and used a botanical semantic resource which enables us to improve access to information contained in a digitized corpus of gardening.

5.1 Assessment Framework of Semantic IR Systems

The task assessed is a search designated ad-hoc in TREC: the IR system responds to an information need with a list of documents arranged in descending order of relevance. The evaluation aims to measure the relative efficiency of our IR system versions named *ThemaStream_i*:

- a classic IR system *Lucene* (baseline) supporting IR based on keywords;
- thematic IR system *ThemaStream₁* supporting semantic IR based on concepts (stage 3, §4);

⁶ <http://moano.liuppa.univ-pau.fr/>

- thematic IR system *ThemaStream₂* supporting semantic IR based on concepts and domain relations (stages 3 and 4, §4).

For a given *topic*, each IR system supplies a list of pairs (d, s) representing the score s of each document returned. We chose the metrics *Mean Relevance Rank* (MRR) and *Precision at 10* (P@10) which, according to [14], correspond to efficient measures for assessing the response quality of an IR system.

Our test collection comprises 25 “topics” (information need) taken from questions asked on the *Yahoo Answers* site, describing information needs in the domain of botany. It involves a “corpus” comprising a sample of 1000 plant cards from the Clause Vilmorin guide, some of which are relevant to the *topics* proposed. The test collection also comprises ontological resources, describing a point of view relating to the botanical domain in the form of concepts and botanical relations.

5.2 Evaluation of the ThemaStream Prototypes

The results shown in Figure 2 confirm the hypothesis that when concepts and relations specifying the content of the need expressed are present in the ontologies brought to bear, a semantic IR approach produces better results than a classic approach based on keywords. Moreover, taking into account domain relations in the query-document matching stage (*ThemaStream₂*) improves consequently the retrieval results. All improvements detected in the results are statistically significant at the 95% confidence interval, according to the t-test.

25 topics	Mean average			
	P@5	P@10	MRR	Number of results
Lucene	0,43	0,46	0,56	405
ThemaStream ₁	0,53	0,58	0,65	910
Improvement//Lucene	23,26%	26,09%	16,07%	
ThemaStream ₂	0,74	0,74	0,83	910
Improvement//Lucene	72,09%	60,87%	48,21%	
Improvement//ThemaStream ₁	39,62%	27,59%	27,69%	

Fig. 2. Overall analysis of results

Indeed, *ThemaStream₁* gives better results than *Lucene* and *ThemaStream₂* gives better results than *ThemaStream₁* and *Lucene*.

Observation of the number of distinct relevant documents in the first ten returned by *Lucene* and *ThemaStream₂* respectively shows the potential complementarity of the two systems. On average, only one relevant document out of the first ten returned is common to *Lucene* and *ThemaStream₂*, all the others being distinct.

6 Conclusion and Prospects

In this paper, we have presented a method of information annotating, indexing and retrieval on the basis of a semantic resource. The evaluation of our approach shows that a semantic IR system based on concepts improves recall with respect to a classic IR system and a semantic IR system based on concepts and semantic relations improves precision with respect to IR based on concepts alone. Thus, when concepts and relations specifying the content of the query are present in the ontological resources, a semantic IR approach produces better results than a classic approach using IR based on keywords.

Although our experiment has 25 topics, the minimum in TREC campaigns, its corpus size is limited to 1000 plant cards. As we focus on domain specific needs and corpora, we may compare it to SemSearch and SemEval campaigns that also make experiments on specific corpora samples. We plan to test our environment with a bigger collection ; the difficulty is to build a domain resource adapted to this collection.

We observe other limits in our experiments. In fact, due to the non-exhaustive nature of the ontology, which does not describe all the plants in the corpus, we detect a bias in the results between classical searching and semantic searching. Thus, Lucene, using keywords, annotates all the documents, whereas our IR system annotates only those documents that contain taxons present in the ontology. Therefore, to remove this ambiguity, two evaluation scenarios need to be set up:

- a measure based on the hypothesis that the ontology describes all the plants in the corpus (the scenario adopted in this paper),
- a measure that eliminates documents annotated by *Lucene* and not annotated by our systems because of the absence of taxons in the ontology.

We plan to work on *ThemaStream* prototypes with a view to enabling a user to construct more expressive queries so as to obtain results refined even further. We would like to set up and evaluate two types of queries:

- either the combination of relations with the operators *AND* or *OR*; for example, *plants that like a maritime climate AND plants that come into bloom in summer*;
- or the extension of a relation by a specifier; for example, *flowers that grow in the shade, in winter*.

Acknowledgements. This research has been conducted in the framework of the “MOANO project (models and tools for nomad applications for discovering territory), financed in part by the Agence Nationale de la Recherche (ANR-2010-CORD-024-01).

References

1. Aussenac-Gilles, N., Kamel, M., Buscaldi, D., Comparot, C.: Construction d'ontologie partir d'une collection de pages web structures. In: Troncy, R. (ed.) Actes des Journées Francophones d'ingénierie des connaissances, IC 2013, Lille, France, pp. 1–16. AFIA (to appear, July 2013)

2. Bannour, I., Zargayouna, H.: Une plate-forme open-source de recherche d'information sémantique. In: CORIA 2012, Bordeaux, France, pp. 167–178 (2012)
3. Cimiano, P., Buitelaar, P., McCrae, J., Sintek, M.: Lexinfo: A declarative model for the lexicon-ontology interface. *Web Semant.* 9(1), 29–51 (2011)
4. Dudognon, D., Hubert, G., Ralalason, B.J.V.: ProxiGénéa: Une mesure de similarité conceptuelle. In: Colloque Veille Stratégique Scientifique et Technologique (VSST), October 2010, Université Paul Sabatier - Toulouse (2010) (support électronique), <http://www.ups-tlse.fr>
5. Dumais, S.T.: Latent semantic analysis. *Annual Review of Information Science and Technology* 38(1), 188–230 (2004)
6. Egozi, O., Markovitch, S., Gabrilovich, E.: Concept-based information retrieval using explicit semantic analysis. *ACM Trans. Inf. Syst.* 29(2), 8:1–8:34 (2011)
7. Fernández, M., Cantador, I., Lopez, V., Vallet, D., Castells, P., Motta, E.: Semantically enhanced information retrieval: An ontology-based approach. *J. Web Sem.* 9(4), 434–452 (2011)
8. Gabrilovich, E.: Feature generation for textual information retrieval using world knowledge. *SIGIR Forum.* 41(2), 123–123 (2007)
9. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pp. 19–33 (1997)
10. 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)
11. Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A.: Kim - a semantic platform for information extraction and retrieval. *Natural Language Engineering* 10(3-4), 375–392 (2004)
12. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)* 11, 95–130 (1999)
13. Roche, C., Calberg-Challot, M., Damas, L., Rouard, P.: Ontoterminology: A new paradigm for terminology. In: *International Conference on Knowledge Engineering and Ontology Development*, Madeira, Portugal, pp. 321–326 (2009)
14. Sanderson, M., Paramita, M.L., Clough, P., Kanoulas, E.: Do user preferences and evaluation measures line up? In: *SIGIR*, pp. 555–562 (2010)
15. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, ACL 1994*, pp. 133–138. Association for Computational Linguistics, Stroudsburg (1994)
16. Zhong, M., Huang, X.: Concept-based biomedical text retrieval. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2006*, pp. 723–724. ACM, New York (2006)

A Latent Semantic Indexing-Based Approach to Determine Similar Clusters in Large-scale Schema Matching

Seham Moawed³, Alsayed Algergawy^{1,2}, Amany Sarhan²,
Ali Eldosouky³, and Gunter Saake¹

¹ Department of Computer Science, Otto-von-Guericke University,
39106 Magdeburg, Germany

² Department of Computer Engineering, Tanta University,
Tanta, Egypt

³ Department of Computer Engineering, Mansoura University,
Mansoura, Egypt

Abstract. Schema matching plays a central role in identifying the semantic correspondences across shared-data applications, such as data integration. Due to the increasing size and the widespread use of XML schemas and different kinds of ontologies, it becomes toughly challenging to cope with large-scale schema matching. Clustering-based matching is a great step towards more significant reduction of the search space and thus improved efficiency. However, methods used to identify similar clusters depend on literally matching terms. To improve this situation, in this paper, a new approach is proposed which uses Latent Semantic Indexing that allows retrieving the conceptual meaning between clusters. The experimental evaluations show encourage results towards building efficient large-scale matching approaches.

Keywords: Large-scale Schema, Clustering-based matching, Similar Clusters, Latent semantic indexing.

1 Introduction

Schema matching is the task of identifying and discovering correspondences between semantically similar elements of two schemas or ontologies [12,14]. The demand of schema matching is great in a diverse number of data application scenarios, such as data integration. Due to heterogeneities inherent in schemas, manual matching becomes expensive, extremely tedious, and error prone. Therefore, efforts are vested in the development of automated schema matching systems. Furthermore, with the rapidly increasing of size and use of XML schemas and ontologies adds another dimensional difficulty of coping with the large matching problem.

To this end, several approaches have been designed to improve the performance of the matching process for large scale schemas involving both aspects; matching effectiveness and efficiency [15,6,7,11,8,13,2]. Among these solutions,

matching techniques that depend on the partition-based principle [6,8,1]. The partition-based matching techniques divide the schema/ontology into a set of partitions and execute a partition wise matching between the two schemas. The partitioning is performed in such a way that each partition of the first schema is matched with only small subset of the partitions of the second schema (ideally, only with one partition) [11]. The entities of the dissimilar partition pairs can be eliminated from further matching process thus reducing the search space to achieve better efficiency. Space complexity of the matching process is also reduced.

To identify partitions, COMA++ uses relatively simple heuristic rules to partition the input schemas resulting often in too few or too many partitions [6]. Both MOM and Falcon are applied only to certain ontology languages and cannot be applied to other data models [15,8]. Algergawy et al. uses a clustering method based on a bottom-up clustering scheme utilizing context-based structural node similarities [1]. To determine similar partitions, COMA++, only uses limited information about the partition (only the root node of the partition) to determine the similarity between partitions of the input schemas. On the other hand, solutions, such as Falcon, fully evaluate the input ontologies to assess the partition similarity. In Algergawy et al. [1], a light-weight similarity measure is applied that considers all elements of each cluster pair and represents each cluster as a cluster document. It uses both of the Vector Space Model and TF-IDF to determine the similarity between cluster documents.

Unfortunately, Vector Space Model depends upon literally matching terms in documents with those of a query [3]. The inaccuracy of lexical matching methods is coming from the inability to determine concepts between documents and the query. So, the literal terms in a user's query may not match those of a relevant document (synonymy). In addition, most words have multiple meanings (polysemy), so terms in a user's query will literally match terms in irrelevant documents. Latent semantic indexing is a more suitable approach that allows retrieving information on the basis of a conceptual topic or meaning of a document [5,9]. To this end, in this paper, we capture features introduced by the Latent semantic indexing technique in large-scale schema matching approaches. To verify the performance of the proposed approach, we conducted a set of experiments in order to prove its superiority upon previous work.

The rest of the paper is structured as follows. We present Latent semantic indexing in Section 2. We then introduce the proposed matching framework in Section 3, concentrating on similar cluster identification. Section 4 reports the experimental results. We conclude in Section 5.

2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a statistical technique, which tries to surpass some limitations imposed by the traditional Vector Space Model (VSM) [5,3]. In VSM, which uses the so-called bag-of-words representation of documents, the collection of text documents is represented by a terms-documents matrix $A = [a_{i,j}] \in R^{t \times d}$, where each entry $a_{i,j}$ corresponds to the number of times the

term i appears in document j . Here t is the number of terms and d is the number of documents in the collection. Therefore a document becomes a column vector and a query of a user can be represented as a vector of the same dimension. The similarity between the user’s query vector and a document vector in the collection is measured as the cosine of the angle between the two vectors. A list of documents ranked in decreasing order of similarity is returned to the user for each query. The VSM considers the terms in documents as being independent from each other, an assumption which is never satisfied by the human language.

Latent Semantic Indexing is a variant of VSM that exploits the dependencies between words by assuming that there is some underlying or "latent" structure in word usage across documents that is partially obscured by variability in word choice and this structure can be revealed statistically. To make the paper self-contained, in the following, main steps of LSI are stated [9]:

- **Constructing Term Document Matrix.** Each term is represented by a row and each document is represented by a column. Initially, each cell in the matrix A , a_{ij} , is represented by the number of times the associated term appears in the indicated document, tf_{ij} . Once the matrix is created, local and global weighting functions can be applied to each non-zero element in the matrix. The weighting functions transform each cell, a_{ij} of A , to be the product of a local term weight which describes the relative frequency of a term in a document, and a global weight, g_i , which describes the relative frequency of the term within the entire collection of documents. The local weighting function of $\log(tf_{ij} + 1)$ decreases the effect of large differences in frequencies. The global weighting function of Entropy which is defined as $1 + \sum_j \frac{P_{ij} \log(P_{ij})}{\log(n)}$ where $P_{ij} = \frac{tf_{ij}}{gf_i}$ is the total number of times the term appears in the entire collection of n documents, gives less weight to terms occurring frequently in a document collection. Therefore, each non-zero element in the term-document matrix is represented as :

$$a_{ij} = (1 + \sum_j \frac{P_{ij} \log(P_{ij})}{\log(n)}) \times \log(tf_{ij} + 1). \tag{1}$$

- **Decomposing The Term Document Matrix.** LSI applies singular value decomposition (SVD) to the matrix A . In SVD, a rectangular matrix is factored into the product of other three matrices as in 2.

$$A = USV^T \tag{2}$$

where $U^T U = I_m$ and $V^T V = I_n$. I_m and I_n are the identity matrices of orders m and n , respectively, and $S_{1,1} \geq S_{2,2} \dots \geq S_{r,r} > 0$, and $S_{i,j} = 0$ where $i \neq j$. Matrix U is an $m \times m$ orthonormal dense matrix and it gives a vector for each term in LSI space, while matrix V is an $n \times n$ orthonormal dense matrix and it represents each document as a vector. S is a diagonal matrix of decreasing singular values. The decomposition given in Eq. 2 actually consumes much more storage space than the original matrix A does.

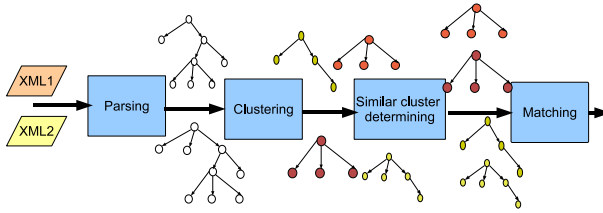


Fig. 1. Schema matching steps

- **Dimensionality Reduction.** LSI computes a low rank approximation to A using a truncated SVD [9]. Let k be an integer and $k \ll \min(m, n)$, U_k is defined to be the first k columns of U , and V_k^T to be the first k rows of V^T . Let $S_k = \text{diag}[s_1, \dots, s_k]$ contain the first k largest singular values as in the following equation:

$$A_k = U_k S_k V_k^T \quad (3)$$

This is a new pseudo term-document matrix with reduced dimension. The SVD operation, along with this reduction, has the effect of preserving the most important semantic information in the text while reducing noise and other undesirable artifacts of the original space of A

- **Incorporating the Query and Ranking the Documents.** A query like a document is a set of words which must be represented as a vector in the k -dimensional space. It can be represented by.

$$q = q^T U_k S_k^{-1} \quad (4)$$

where q is the vector of words in the users query, multiplied by the appropriate term weights. The query vector can then be compared to all existing document vectors, and the documents ranked by their similarity (nearness) to the query. One common measure of similarity is the cosine between the query vector and document vector. Typically, the z closest documents or all documents exceeding some cosine threshold are returned to the user.

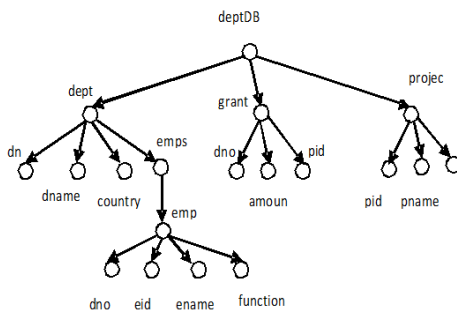
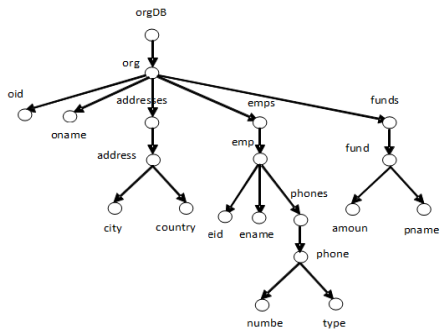
3 The Matching Framework

The framework of the proposed schema matching approach presented in this paper consists of five consecutive stages, as shown in Fig. 1.

3.1 Schema Parsing

XML schemas are first parsed using the XML Schema Object Model (XSOM) parser¹. XSOM is a Java library that allows applications to easily parse XML Schema documents and inspect information in them. The library is a straightforward implement of "schema components" as defined in the XML Schema.

¹ <https://xsom.java.net>


Fig. 2. Schema graph, deptDB

Fig. 3. Schema graph, orgDB

3.2 Schema Preparation

To make the matching process a generic process, schemas to be matched should be represented internally by a common representation. In our implementation, we internally represent XML schemas using graph data structure, called *schema graph*. Figs 2 and 3 represent the schema graph representation of two XML schemas taken from [4]. Both DeptDB and orgDB represent information about departments with their employees and grants, as well as the projects for which grants are awarded.

3.3 Schemas Clustering

We make use of our clustering algorithm presented in [1]. To make the paper self-contained, we present a brief detail about the algorithm. After internally representing each XML schema as a schema graph, the algorithm is devoted to cluster each schema graph into a set of disjoint sub-graphs, such that nodes in the same cluster are structurally similar. To this end, we compute the structure similarity between every pair of nodes in the schema graph based on their contexts. The context of a node, $C(v_i)$ is the combination of the node itself as well as all parents and children of the node.

The structure similarity between two nodes v_i and v_j which exist in the same SG is computed based on the number of common nodes between their contexts, $|C(v_i) \cap C(v_j)|$. Based on this structural similarity, we construct a link between each node pair, containing the two nodes and their structural similarity. The set of generated links constitutes a has table called *Links hash table*. This table is used as an input for the clustering algorithm.

Example 1. Applying the clustering algorithm to two schema graphs illustrated in Figs. 2 & 3, we get two cluster sets. $CSet_1 = \{C_{11}, C_{12}\}$ and $CSet_2 = \{C_{21}\}$ for deptDB and orgDB schemas, respectively.

3.4 Similar Cluster Determination

In this paper, the focus of the proposed approach is on 2-way or pairwise schema matching where two related input schemas are matched with each other. From the clustering layer, the result is a pair of a set of clusters; each constitutes the best level of its own schema graph and contains a set of clusters; each contains some nodes that are structurally similar.

Latent semantic indexing aims to detect semantically similar partitions (clusters) in the two schema graphs. This motivation reduces the match overhead by applying matching only on similar partitions and ignoring the irrelevant ones. Algorithm 1 is proposed to achieve this task. It accepts two sets of clusters as an input and processes to determine similar clusters across the two sets. The algorithm has the following main steps, as shown in Algorithm 1.

Algorithm 1. Similar clustering determination

Require: Two sets of clusters, $CSet_1 = \{C_{11}, C_{12}, \dots, C_{1n}\}$ and $CSet_2 = \{C_{21}, C_{22}, \dots, C_{2m}\}$
Ensure: A set of similar clusters, $Sim_Clust = \{(C_{1i}, C_{2j}) | C_{1i} \in CSet_1, C_{2j} \in CSet_2\}$
 { // Step 1: Preparation }
 1: $A \leftarrow analysis(CSet_1)$;
 2: Compute for each entry in $A : a_{ij}$
 $a_{ij} \leftarrow (1 + \sum_j \frac{P_{ij} \log(P_{ij})}{\log(n)}) \times \log(tf_{ij} + 1)$
 { // Step 2: Singular Value Decomposition & reduction }
 3: Apply SVD to $A : A = USV^T$
 4: Dimensionality reduction: $A_k = U_k S_k V_k^T$
 { // Step 3: Query incorporating and folding }
 5: $q \leftarrow analysis(CSet_2)$;
 6: $q \leftarrow q^T U_k S_k^{-1}$;
 { // Step 4: Similarity calculating and ranking }
 7: $sim(q, \leftarrow q^T U_k S_k^{-1})$;
 8: **for** $column \in A$ **do**
 9: $d_i \leftarrow A$;
 10: **for** $cluster \in CSet_2$ **do**
 11: $q_j \leftarrow form(C_{2j})$;
 12: $simMat[i][j] = sim(q_j, d_i)$;
 13: **end for**
 14: **end for**

- Stage 1: Preparation of term-document matrix. First, all elements in the first cluster set ($CSet_1$) are extracted and analyzed. A set of normalization process has been applied to the element names, such as tokenization, stemming in order to obtain non-repeating terms in the cluster set. So that, we have created the terms vector by collecting the names of the nodes. As a next step, the term-document matrix is initially created with each matrix cell representing the number of times the associated node name appears in the

indicated clusterdocument, line 1. Finally, we apply log-entropy weighting function to our matrix, line 2. Thus, the term-document matrix is ready for the next stage.

- Stage 2: Applying singular value decomposition. To construct a semantic space where in the names of the nodes in the $CSet_1$ and clusters that are closely associated are placed near one another, we apply the singular decomposition value technique. The technique factorizes a term-document matrix into its left singular vectors, right singular vectors, and singular values, line 3. Each node name within the cluster set is now represented by a singular vector via matrix U. Additionally, each cluster is represented by a singular vector via matrix V.
- Stage 3: Reduction. To reduce the noise and redundancy, LSI uses a truncated SVD, line5, which consists in retaining only the largest k singular values and deleting the remaining ones which are smaller and thus considered unimportant. The columns corresponding to the small singular values are also removed from U and V. So, SVD allows the arrangement of the space to reflect the major associative patterns in the data, and ignore the smaller, less important influences. As a result, terms that do not actually appear in a document may still up close to the document, if that is consistent with the major patterns of association in the data.
- Stage 4: Folding. The following step is to prepare set of clusters in the second cluster set $CSet_2$. Each cluster is treated as a user query. First, we analyze the element names of each cluster and we apply a set of normalization process to these elements. The query is then treated as an ordinary document and hence it should be put with new coordinates in the reduced k-dimensional space, lines 5&6.
- Stage 5: Calculating similarities and ranking the document. Now, the documents are represented via matrix V and also through the folding in process, the query is represented into the new reduced semantic dimensional space. A latter step, in order to get the similarity between the document vector and the query, we use cosine similarity function between two vectors, line 12.

The computed similarities between cluster pairs of the two schemas are used to construct a so-called cluster similarity matrix. The elements of the matrix are ranked according to their similarity to each other and the *top - k* elements are selected from the ranked list.

Once settling on the similar clusters of the two schemas, the next step is to fully match similar clusters to obtain the correspondences between their elements. Each pair of the similar clusters represents an individual match task that is independently solved. Since the main scope of this paper is to identify similar clusters, we do not go through more discussion of this section.

3.5 Walk-Through Example

We provide an example that describes the proposed method and determines the similarity score. In this example, we use two schema graphs illustrated in Figs. 2 & 3. We formulate the problem in this example as follows: given two cluster

sets $CSet_1 = \{C_{11}, C_{12}\}$ and $CSet_2 = \{C_{21}\}$, identify similar clusters across the two cluster sets.

- Steps 1: Constructing term-document similarity and applying log-entropy weighting function, we get the following term-document matrix, A

$$\begin{matrix}
 \begin{bmatrix}
 dept \\
 DB \\
 dno \\
 dname \\
 country \\
 emps \\
 emp \\
 eid \\
 ename \\
 function \\
 grant \\
 amount \\
 pid \\
 project \\
 pname \\
 year
 \end{bmatrix} & A_i = & \begin{bmatrix}
 1 & 1 \\
 0 & 1 \\
 2 & 1 \\
 1 & 0 \\
 1 & 0 \\
 1 & 0 \\
 1 & 0 \\
 1 & 0 \\
 1 & 0 \\
 1 & 0 \\
 1 & 0 \\
 0 & 1 \\
 0 & 1 \\
 0 & 2 \\
 0 & 1 \\
 0 & 1 \\
 0 & 1
 \end{bmatrix} & \Rightarrow & A_{entropy} = & \begin{bmatrix}
 0 & 0 \\
 0 & 0.69 \\
 0.09 & 0.06 \\
 0.69 & 0 \\
 0.69 & 0 \\
 0.69 & 0 \\
 0.69 & 0 \\
 0.69 & 0 \\
 0.69 & 0 \\
 0.69 & 0 \\
 0.69 & 0 \\
 0 & 0.69 \\
 0 & 0.69 \\
 0 & 1.1 \\
 0 & 0.69 \\
 0 & 0.69 \\
 0 & 0.69
 \end{bmatrix}
 \end{matrix}$$

- Step 2: the log-entropy matrix is then decomposed and its dimensions are truncated. For truncation, we assume to truncate 98% of the singular values. In this example, there is no truncation.

$$V = \begin{bmatrix} -0.007 & -1 \\ -1 & 0.007 \end{bmatrix} \quad S = \begin{bmatrix} 2.02 & 0 \\ 0 & 1.84 \end{bmatrix}$$

- Step 3: Incorporating query, we incorporate the clusters of orgDB schema into the new dimensional space created by SVD and its reducing form processes. The schema is partitioned into one cluster according to the applied threshold. Hence, we have one query. The new coordinates of this query is represented in vector q , $q = [-0.131 - 0.262]$.

- Step 4: The final step is applying cosine similarity function and ranking the documents as follows.

$sim(C_{11}, C_{21}) = sim(d_1, q) = 0.897$ and $sim(C_{12}, C_{21}) = sim(d_2, q) = 0.442$
 It should be noted that we solve the same example using VSM and we get the following results: $sim_{VSM}(C_{11}, C_{21}) = 0.373$ and $sim_{VSM}(C_{12}, C_{21}) = 0.224$. From this example, it has been shown that the computed similarities by LSI are better than those computed by VSM due to the ability of LSI to correlate semantically related terms that are latent in the collection of documents. The documents, among them query vectors, are represented with new dimensions with semantic correlation between them.

Table 1. Data set specification

Domain	Tested sources	No. of elements
Spicy	deptDB/orgDB	19/20
University	Uni1/Uni2	11/11
Web	Yahoo/ebay	37/37
TPC_H	TPC_H1/TPC_H2	43/17
Finance	finan1/finan2	14/14
GeneX	GeneX1/GeneX2	75/85
Mondial	Mondial1/Mondial2	117/108
PO(large)	OpenTran_Invoice/OpenTran_Order	1113/1162

Table 2. Results

Domain	No. of clusters	No. of similar cluster	
		VSN-based	Latent-based
Spicy	2/1	2/2	2/2
University	1/2	2/2	2/2
Web	4/3	4/6	4/6
TPC_H	6/1	1/2	2/2
Finance	1/2	2/2	2/2
GeneX	10/8	10/15	10/16
Mondial	10/10	8/20	10/20
PO(large)	57/56	80/112	100/112

4 Experimental Evaluation

To evaluate the effectiveness of the proposed approach, we conducted a set of experiments utilizing real-world schemas and ontologies of different sizes [6,10]. Table 1 shows the characteristics of the test schemas from different domains². More details about data sets in Table 1 can be found in [6,10]. We ran all our experiments on 2.67GHz Intel (R) Core i5 processor with 4GB RAM running Windows 7. We implemented the approach in Java.

4.1 Experimental Results

We validated the proposed approach using XML schemas illustrated in Table 1. Each XML schema is parsed and represented as a schema graph. The clustering-based approach, in [1], is applied to partition each schema graph into a set of clusters. To determine similar clusters among two sets of clusters, we applied both our latent semantic-based approach and the VSM-based approach [1]. The elements of cluster similarity matrix are ranked according to their similarity to each other and the top-2 are selected. Results are summarized in Table 2.

Table 2 shows that both LSI-based and VSM-based approaches produce the same results for small-scale schemas, however, the LSI-based approach outperforms the VSM-based approach with respect to medium and large schemas. Furthermore, from the obtained results, the LSI-based approach produces very lower similarity values when comparing dissimilar clusters than the VSM-based approach across different domains. This means that if we apply a threshold-based selection (instead of top-2), the LSI-based approach is more effective than the VSM-based approach.

5 Conclusions

Current schema matching approaches still have to improve for large and complex schemas. Identifying similar partitions of two schema graphs is a crucial step before the matching process. To this end, we proposed a new approach for

² <http://queens.db.toronto.edu/project/clio/index.php#testschemas>

detecting similar clusters; latent semantic indexing (LSI); an indexing and retrieval method which had been proven in the literature to be a useful solution to a number of conceptual matching problems. The proposed approach is compared against classical vector space (VSM) in the scope of large-scale schema matching. To verify the performance of the proposed approach, we conducted a set of experiments. From the results, it is deduced that LSI outperforms VSM in detecting the most similar clusters and gives suitable similarity values for each pair. In future work we extend the framework to explore the effect of LSI on matching performance aspects.

References

1. Algergawy, A., Massmann, S., Rahm, E.: A clustering-based approach for large-scale ontology matching. In: Eder, J., Bielikova, M., Tjoa, A.M. (eds.) ADBIS 2011. LNCS, vol. 6909, pp. 415–428. Springer, Heidelberg (2011)
2. Algergawy, A., Schallehn, E., Saake, G.: Improving XML schema matching using pruner sequences. *DKE* 68(8), 728–747 (2009)
3. Berry, M.W., Drmac, Z., Jessup, E.R.: Matrices, vector spaces, and information retrieval. *SIAM Review* 41(2), 335–362 (1999)
4. Bonifati, A., Mecca, G., Pappalardo, A., Raunich, S., Summa, G.: Schema mapping verification: the spicy way. In: EDBT 2008, France., pp. 85–96 (2008)
5. Deerwester, S., Dumais, S.T., Harshman, R.: Indexing by latent semantic analysis. *Journal of American Society for Information Science* 41, 391–407
6. Do, H.H., Rahm, E.: Matching large schemas: Approaches and evaluation. *Information Systems* 32(6), 857–885 (2007)
7. Hamdi, F., Safar, B., Reynaud, C., Zargayouna, H.: Alignment-based partitioning of large-scale ontologies. In: Guillet, F., Ritschard, G., Zighed, D.A., Briand, H. (eds.) *Advances in Knowledge Discovery and Management. SCI*, vol. 292, pp. 251–269. Springer, Heidelberg (2010)
8. Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: A divide-and-conquer approach. *DKE* 67, 140–160 (2008)
9. Landauer, T.: *Handbook of Latent Semantic Analysis* (2007)
10. Peukert, E., Massmann, S., König, K.: Comparing similarity combination methods for schema matching. In: *GI-Workshop*, pp. 692–701 (2010)
11. Rahm, E.: Towards large-scale schema and ontology matching. In: *Data-Centric Systems and Applications*, vol. 5258, pp. 3–27. Springer (2011)
12. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. *VLDB Journal* 10(4), 334–350 (2001)
13. Seddiquia, M.H., Aono, M.: An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics* 7(4), 344–356 (2009)
14. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.* 25(1), 158–176 (2013)
15. Wang, Z., Wang, Y., Zhang, S.-S., Shen, G., Du, T.: Matching large scale ontology effectively. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) *ASWC 2006. LNCS*, vol. 4185, pp. 99–105. Springer, Heidelberg (2006)

Poss – *SRIOQ(D)*: Possibilistic Description Logic Extension toward an Uncertain Geographic Ontology

Safia Bal Bourai¹, Aicha Mokhtari², and Faiza Khellaf²

¹ Ecole nationale Supérieure d'Informatique,
BP 68M 16309, Oued-Smar, Alger, Algeria
s_bourai@esi.dz

² RIIMA, Computer Science Department, USTHB, Algiers, Algeria
Aissani_Mokhtari@yahoo.fr, fkhellaf@usthb.dz

Abstract. The use of description logics (DL) formalism to represent geographical knowledge has received a lot of attention recently. Nevertheless, classical DLs are not suitable to represent incomplete and uncertain knowledge, which represent several situations in geographic domain. In addition they cannot represent the spatio-temporal information usually present in geographical application. In this paper, we propose a possibilistic extension of the very expressive Description Logic *SRIOQ(D)*, the basis of the language OWL2, called *Poss*–*SRIOQ(D)*, as a solution to handling uncertainty and for dealing with inconsistency in geographical applications. Both syntax and semantics of *Poss* – *SRIOQ(D)* are considered. Illustrative examples are given.

Keywords: Description Logics, ontology, GIS, uncertainty, possibilistic logic.

1 Introduction

Spatial information has become so common today that they are available from each personal GPS in smart phone or smart car. At the same time, Geographic Information Systems (GIS) developments and integrations lead to a variety of geographic applications such as environmental, cadastral or touristic one. The spatial data are an approximation of reality. They can be localized in space according to their different degrees of precision and dated with more or less precision and different levels of temporal granularity. This leads us to ask about the structure of the spatial data in their different dimensions, namely: spatial and temporal attribute under uncertainty.

Indeed, uncertainty is the inability to accurately specify something. Obviously, it is in the interest of users and decision makers to locate the known uncertainties in data, and must be aware of the serious consequences that can result from overly precise geographic information.

Geographic ontologies are widely studied and a number of interesting research challenges have been reached but, a little works have been done on the uncertain aspect of these ontologies.

Our aim, in this paper, is to address this aspect by taking advantage of research results in uncertain knowledge representation especially in possibilistic logic. The main objective is to develop a solution to represent uncertainty in existing geographic data in terms of concepts and their relationships in order to reduce the gap between real data and its utilization in geographic applications. For this purpose we define a qualitative possibilistic extension of $SRIOQ(\mathcal{D})$ description logic called $Poss-SRIOQ(\mathcal{D})$. The choice of $SRIOQ(\mathcal{D})$, the basis of OWL2 language, is essentially related to its ability to represent the spatio-temporal information usually present in geographical application.

The paper is organized as follows. Section 2 provides the main types of uncertainty in geographic information. Section 3 is dedicated to the possibilistic extension of the description logic $SRIOQ(\mathcal{D})$. Section 4 presents related work. Finally, section 5 concludes and presents some future works.

2 Uncertainties in Geographic Information

Geographic information systems represent objects or phenomena located on space and time (such as roads, constructions, elevations, pollution, etc). The particularity of spatial objects or phenomena is precisely their relationship to space, which they are intrinsically linked [16]. Therefore, representing these concepts requires specific elements, such as : location, shape, orientation, relations between objects, etc.

GIS can organize objects and phenomena in space, but it cannot, for example, determine whether a statement is true or false or infer new information.

Ontologies have a growing interest in geographic information processing field. Spatial ontologies [6], [19] represent geographical concepts that describe the geographic area or entities or phenomena of the space. Geographic ontologies can help to understand spatio-temporal objects and their relationships, because they are intentionally used to define spatial concepts with axioms and to reasoning about instances to infer valid relationships from existing ones. Finally, they appear as an essential component to achieve integration of heterogeneous data sources.

Uncertainty exists in every phase of the life cycle of geographic object (data collection, data representation, data analyses and final results). Uncertainty can result from a lack of information [20]. Bouchon-Meunier [3] present two sources of imperfection:

- Observation and representation: Observation occurs by instruments or humans. Representations are expressed in natural language, the numbers are given with a certain precision or a mathematic formula.
- Lack of rigorous and flexibility in the system itself and its operations: This is the case for all the characteristics of natural phenomena and some artificial systems.

Uncertainty of geographic information may be derived from measurement errors, registration, classification, clustering classes, generalization and temporal processing [9].

Shu and AL. [16] distinguish two main types of geographic information uncertainty :

- Uncertain spatio-temporal data type, which can be modeled by the uncertainty of its thematic, spatio-temporal (point, line and zone) and time (instant and interval) attributes.
- Uncertain spatio-temporal relationships, refers to the topological spatial relationships [5] and topological temporal relationships [1].

There are several forms of uncertainty in spatial domain. For us, two main types of uncertainty must be distinguish: spatio-temporal object uncertainty and relationships uncertainty.

1- Spatio-temporal object uncertainty, which refers to the uncertainty of the thematic, spatial and temporal attributes, such as: name, size, date, duration, position, etc. This form is caused by: data dated without updating, data from inaccurate calculation, error type data, imprecise data, several versions of the same object, missing data, lost historical data, etc.

2- Relationships uncertainty, which is divided into the following forms:

a) Spatio-temporal relationships uncertainty, describes the spatio-temporel relationships between objects. These reflect spatial relationships associated to objects, such as: (disjoint, touch, overlap, equal, cover, contain, contained by, disconnected from,...) and temporal relationships between intervals of time or durations associated to objects, such as: before, during, meets, etc. This form of uncertainty can be caused by an incomprehension or by a lack of clarity in reality.

b) Subsumption relationships or spatio-temporal objects classification uncertainty, reflects the relations among concepts (set of individuals or objects), such as: is-a, kind-of, has-part, etc. This form of uncertainty is caused by: uncertain labels, different data models, multiple meanings for definition of an object or a relationship, overlap between definitions of objects or disagreement on definition of others, etc.

c) Belonging relationships uncertainty, refers to the attribution of data to the corresponding concept. This form is caused by: lack of informations, multiple meanings for definition of an object, etc.

3 *Poss – SROIQ(D)*: Possibilistic Extension of *SROIQ(D)*

In the previous section, we highlighted two levels relative to several situations characterized by some forms of uncertainty: spatio-temporal object uncertainty level and relationships level. In order to deal with these uncertainties, we propose, in this section, possibilistic extension of the description logic *SROIQ(D)*[12]. Possibilistic logic [7] provides an efficient solution for handling uncertain or prioritized formulas and coping with inconsistency. It is particularly suitable for

the representation of states of partial or complete ignorance. Possibilistic logic is developed in two directions: qualitative and quantitative.

In the qualitative direction, the possibility measure allows the evaluation of the knowledge by using the order notion. Knowledge are organized on stages (strates) according to their degrees of incertitude and formulas of the same degree occupy the same strate. The qualitative possibility logic is presented as a set of postulates that constrain any reasonable ordering of possibility on sentences. It offers sufficient expressive power to capture the types of constraints that might be required in a knowledge base.

Naturally, geographic knowledges appear qualitative. To reduce their uncertainty, we focus on the qualitative possibilistic approach to represent uncertainty related to geographical objects and their relationships. Let us note that in the literature there are little works in this topic. Both syntax and semantic of $\mathcal{P}oss - \mathcal{S}ROIQ(\mathcal{D})$ are described.

3.1 $\mathcal{P}oss - \mathcal{S}ROIQ(\mathcal{D})$ Syntax

$\mathcal{P}oss - \mathcal{S}ROIQ(\mathcal{D})$ is a combinaison of both DL and possibilistic logic. Alphabet of symbols are proposed: possibilistic concepts, possibilistic roles and possibilistic individuals. Possibilistic concepts denote possibilistic sets of individuals and possibilistic roles denote possibilistic relationships.

Let C, D be concepts, A an atomic concept, R an abstract role, S a simple role, $T, T1$ and $T2$ be concrete roles, d a concrete predicate, a, b an abstract individuals, v a concrete individual, n a natural numbers with $n \geq 0$ and $\alpha \in (0, 1]$.

A concrete domain (Δ_D, ϕ_D) , where Δ_D is an interpretation domain and ϕ_D is a set of concrete predicates d on the domain Δ_D with interpretation

$$p^D : \Delta_D^n \mapsto [0, 1]$$

The concepts can be built using the following rule:

$$C, D \rightarrow \top \mid \perp \mid A \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \forall R.C \mid \exists R.C \mid \forall T.d \mid \exists T.d \mid \geq nS.C \mid \leq nS.C \mid \geq nT.d \mid \leq nT.d \mid \exists S.Self \mid \{(o_i, \alpha_i)\} \mid (A, \alpha).$$

The roles can be built according to the following rule:

$$R \rightarrow R_A \mid R^- \mid U.$$

The difference with the non-possibilistic case is the presence of weighted nominals and weighted concepts.

- Weighted nominals are of the form: $\{(o_i, \alpha_i)\} (o_i \neq o_j, 1 \leq i < j \leq n)$. They represent individual uncertainties.
- Weighted concepts are of the form: (A, α) . They define a set of individual uncertainties.

Let us illustrate this new definition of weighted concept using examples of geographic informations from archaeological domain.

Example 1. Let consider the concept of Historic-area represented by: $\{(Tipaza, 0.9), (Blida, 0.2), (Tlemcen, 0.8)\}$, where Tipaza, Blida and Tlemcen are individuals belonging to Historic-area at least to the degree 0.9, 0.2 and 0.8 respectively.

The set of Historic-areas with values at least to the degree 0.8 is: $\{(Tipaza, 0.9), (Tlemcen, 0.8)\}$.

Possibilistic knowledge base Σ is a finite set of weighted axioms in the form (φ, α) . The axioms consist of a possibilistic TBox denoted by PossTBox and possibilistic ABox denoted by PossABox.

1. PossTBox: PossTBox consists of a finite set of possibilistic GCIs (General Concept Inclusions) and a finite set of role axioms noted, RIAs (Role Inclusion Axioms), expressed by the following forms:
 - $(C \sqsubseteq D, \alpha)$, where C and D are concepts and $\alpha \in (0, 1]$. $(C \sqsubseteq D, \alpha)$ expresses that $C \sqsubseteq D$ is certain at least to the degree α .
 - $(w \sqsubseteq R, \alpha)$, where w is a role chain, expresses that $w \sqsubseteq R$ is certain at least to the degree α .
 - $(T_1 \sqsubseteq T_2, \alpha)$, expresses that $T_1 \sqsubseteq T_2$ is certain at least to the degree α .

Example 2. Let us consider the concretes roles haslocalization and hasaddress. The possibilistic RIA: haslocalization \sqsubseteq hasaddress, 0.3 states that haslocalization can also be considered as an address at least degree 0.3.

Further, we can add: Transitive role axiom Trans(R), Disjoint role axiom $Dis(S_1, S_2)$, $Dis(T_1, T_2)$ Reflexive role axiom Ref(R),...as defined in *SRQIQ(D)*.

Possibilistic TBox allows to express the uncertainty of subsumption relationships or spatio-temporal objects classification and subsumption Roles.

2. PossABox: PossABox consists of a finite set of possibilistic assertions expressed in the following forms: $(C(a), \alpha)$, $(R(a, b), \alpha)$ or $(\neg R(a, b), \alpha)$, $(a = b, \alpha)$ or $(a \neq b, \alpha)$, $T((a, v), \alpha)$ or $\neg T((a, v), \alpha)$.
 - $(C(a), \alpha)$: It means that the individual is certainly in the concept C at least with the degree α^1 .

Example 3. Let us consider the possibilistic assertion: tombeau de la chrétienne: Historic-monument, 0.75, where tombeau de la chrétienne is an individual and Historic-monument is a concept, which states that tombeau de la chrétienne is a Historic-monument with at least degree 0.75.

- $(R(a, b), \alpha)$ or $(\neg R(a, b), \alpha)$: The first possibilistic role assertion means that it is certain that the individuals a and b are related by the abstract role R at least with the degree α . The second means that it is certain that the individual a and b are not related by the abstract role R at least with the degree α^2 .

¹ This form allows to express the type (c) of uncertainty in spatial domain, which is belonging relationships uncertainty.

² These forms allow to express the type (a) of uncertainty in spatial domain, corresponding to the spatio-temporal relationships uncertainty.

Example 4. Let us consider two objects "phare street" and "Tipaza littoral", which have spatial relationships between each others. The possibilistic assertion: $\text{close}((\text{phare street}, \text{Tipaza littoral}), 0.9)$ states that "phare street" and "Tipaza littoral" are closed at least with the degree 0.9. In another hand the possibilistic assertion: $\neg \text{closed}((\text{phare street}, \text{Tipaza littoral}), 0.5)$ states that "phare street" is far from "Tipaza littoral".

- $(a = b, \alpha)$ or $(a \neq b, \alpha)$: The first possibilistic assertion means that it is certain at least with the degree α that a and b represent the same individual. The second means that it is certain at least with the degree α that a and b represent two different individuals. These forms are considered as specific cases of the two precedent assertions, where R is designed by the relation "equal" (=).
- $T((a, v), \alpha)$ or $\neg T((a, v), \alpha)$: The first possibilistic data types role means that individual a has certainty the value v for the datatype role T at least with the degree α . The second means that individual a has certainty the value v for the datatype role $\neg T$ at least with the degree α^3 .

3.2 Poss – SROIQ(D) Semantics

The semantic level is based on the notion of a possibility distribution, which is a mapping from a set of interpretations to the interval $[0,1]$. For each possibilistic knowledge base, there is a unique possibility distribution associated with it. The possibility distribution of an interpretation I, denoted $\pi(I)$, is the degree of compatibility of interpretation I with available beliefs. From a possibility distribution, two important measures can be processed:

- The possibility degree of a formula φ , defined as $\Pi(\alpha) = \max\{\pi(I), I \models \varphi\}$
- The certainty degree of a formula φ , defined as $N(\varphi) = 1 - \Pi(\neg\varphi)$.

An interpretation $I = (\Delta^I, I)$ consists of a non empty set Δ^I , called the domain of I, and a valuation I which associates, with each concept C a function $C^I : \Delta^I \rightarrow [0, 1]$, with each abstract role R, a function $R^I : \Delta^I \times \Delta^I \rightarrow [0, 1]$, with each individual a, an element $a^I \in \Delta^I$, with each concrete individual v, an element $v_D \times \Delta_D$, with each concrete role T a function $T^I : \Delta^I \times \Delta_D \rightarrow [0, 1]$ and to each n-ary concrete predicate d the interpretation $d_D : \Delta_n^D \rightarrow [0, 1]$.

Table 3 and Table 4 show respectively the semantics of concepts and roles and axioms of $\text{Poss} - \text{SROIQ}(\mathcal{D})$. Let us note that for the semantics, we are inspired by the combination functions used in the context of Zadeh logic [20] and [2].

Example 5. Let us the following relationships:

- The concept (Exist \sqcup Accessible) describes the existing streets or Accessible ones. The degree of uncertainty that the street du Phare yet exist is 0.9 or the degree of uncertainty that this street is accessible is 0,5, then the certainty that "street du Phare" is both yet exist or Accessible is 0.9.

³ These forms allow to express the type (1) of uncertainty in spatial domain, which is spatio-temporel object uncertainty.

Table 1. Interpretation of concepts and roles in $\mathcal{P}_{oss} - \mathcal{SROIQ}(\mathcal{D})$

Concept	Semantics
\top	1
\perp	0
$C \sqcap D$	$\inf(C^I(x), D^I(x))$
$C \sqcup D$	$\sup(C^I(x), D^I(x))$
$\neg C$	$1 - C^I(x)$
$\forall R.C$	$\inf_{y \in \Delta^I} (R^I(x, y) \Rightarrow C^I(y))$
$\exists R.C$	$\sup_{y \in \Delta^I} (\inf (R^I(x, y), C^I(y)))$
$\forall T.d$	$\inf_{v \in \Delta D} (T^I(x, v) \Rightarrow d_D(v))$
$\exists T.d$	$\sup_{v \in \Delta D} (\inf (T^I(x, v), d_D(v)))$
$\geq_n \text{ S.C}$	$\sup_{y_i \in \Delta^I} (\inf_{i=1}^n (S^I(x, y_i), C^I(y_i)))$ and $(y_i = y_j)$
$\leq_n \text{ S.C}$	$\inf_{y_i \in \Delta^I} (\sup_{i=1}^{n+1} (S^I(x, y_i), C^I(y_i)) \Rightarrow y_i \neq y_j)$
$\geq_n \text{ T.d}$	$\sup_{v_i \in \Delta D} (\inf_{i=1}^n (T^I(x, v_i), d_D(v_i)))$ and $(v_i = v_j)$
$\leq_n \text{ T.d}$	$\inf_{v_i \in \Delta D} (\sup_{i=1}^{n+1} (T^I(x, v_i), d_D(v_i)) \Rightarrow v_i \neq v_j)$
$\exists \text{ S.Self}$	$S^I(a, a)$
$\{(o_1, \alpha_1), \dots, (o_n, \alpha_n)\}$	$\sup(\alpha_i)^I$
(A, α)	$A^I(a)$
Role	Semantics
R_A	$R^I(a, b)$
R^-	$R^I(b, a)$
U	1

- A set of precipitation period from Mars until May with the degree uncertainty of 0.5, 0.3, 0.3 respectively. This situation can be represented by the set : $\{(Mars, 0, 5), (April, 0, 3), (May, 0, 3)\}$, then the certainty of the precipitation period is 0.5.

Table 2. Interpretation of axioms in $\mathcal{P}_{oss} - \mathcal{SROIQ}(\mathcal{D})$

Axioms	Semantics
$(C \sqsubseteq D)^I$	$\inf_{x \in \Delta} C^I(x) \Rightarrow D^I(x)$
$(R_1, \dots, R_m \sqsubseteq R)^I$	$\inf_{x_1, x_{n+1} \in \Delta} I(\sup(\inf(R_1^I(x_1, x_2), \dots, R_n^I(x_n, x_{n+1}))) \Rightarrow R_n^I(x_1, x_{n+1}))$
$(T_1 \sqsubseteq T_2)^I$	$\inf_{x \in \Delta^I, v \in \Delta D} T_1^I(x) \Rightarrow T_2^I(v)$
$(a : C)^I$	$C^I(a^I)$
$((a, b) : R)^I$	$R^I(a^I, b^I)$
$((a, b) : \neg R)^I$	$1 - R^I(a^I, b^I)$
$(a = b)^T$	$a^I = b^I$
$(a \neq b)^T$	$a^I \neq b^I$
$((a, v) : T)^I$	$T^I(a^I, v_D)$
$((a, v) : \neg T)^I$	$1 - T^I(a^I, v_D)$

4 Related Work

A few approaches for managing uncertainty in geographic information have been proposed recently. Inspired by the theories of stochastic, fuzzy, methods of prob-

abilistic and statistic databases, Shu and AL.[16] suggest an uncertainty model including uncertain spatio-temporal data types and spatio-temporal relationships. Pfoser and AL. [14] present probabilistic models to model position and attributes errors.

The probabilistic model is still the most used model. Nevertheless, it presents some limits. For example, it don't make distinct between the ignorance and the uncertainty, and it don't allow the representation of total ignorance. The possibilistic logic is particularly suitable for the representation of states of partial or complete ignorance. In the qualitative direction, the possibility measure permit the evaluation of the knowledge by using the order notion, where Knowledges are organized on stages according to their degrees of incertitude. So, we focus on the qualitative possibilistic approach to represent uncertainty related to geographic objects and their relationships.

Dupin [8] develops logical framework in a possibilistic approach for handling uncertain spatial information called attributive formula, and merging it when it comes from multiple Source. Attributive formula is a pair made of a property and a set of parcels (to which the property applies). The notion of spatial relationships is not considered.

The relevant works related to our solution are [2],[15]. Bobillo and Straccia [2] present a fuzzy version of *SRIOQ* and provide a reasoning capabilities of fuzzy *SRIOQ*. This work is more adapted for fuzzy information like : tall, old, large,...than uncertain one. Qi[15] propose a possibilistic description logic as an extension of description logic. The syntax of description language is the same as the standard DL and the interpretation is based on possibility theory. This solution is then not adapted for managing the various forms of geographic uncertainty such as the type 1, corresponding to the uncertainty of spatio-temporal object, mentioned, in section 3 .

Our approach differs from the existing ones on extending the description logic *SRIOQ(D)* by possibilistic logic, where concepts, individuals and axioms are extended to the possibilistic case. It allows the representation of several forms of geographic uncertainty at the spatio-temporel objects level and relationships level.

5 Conclusion

In this paper, we have considered the problem of uncertainty in GIS. First, two main categories of uncertainty were identified according to the taxonomy of geographic information uncertainty, namely, spatio-temporal object uncertainty and relationships uncertainty. Then, we have proposed a solution to deal with these uncertainties.

In this solution, we have proposed a possibilistic extension of the *SRIOQ(D)* called *Poss - SRIOQ(D)*. This allows us to consider geographic objects and phenomena, described by, the uncertainty of their concepts, individuals, attributes and relationships. We have presented the syntax and the semantic of *Poss - SRIOQ(D)* illustrated by geographic information examples.

Our future work includes inference problems in *Poss* – *SROIQ*(\mathcal{D}) logics by developing a reasoner and testing its usefulness for other applications. Some free tools exist, in the web, for reasoning on description logic or possibilistic one. We should probably use them by means of some modifications. However, currently we can not compare our proposition to others because of the nonexistence of similar approaches. Indeed geographic ontologies can help to represent spatio-temporal objects and their relationships and to infer valid relationships from existing ones. They are formulated in ontology languages. So, another proposal corresponding to a possibilistic ontological language, based on our DL *Poss* – *SROIQ*(\mathcal{D}), will be considered. It corresponds to an extension of the OWL2.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11), 832–843 (1983)
2. Bobillo, F., Straccia, U.: Reasoning with the finitely many-valued Lukasiewicz fuzzy Description Logic SROIQ. *Information Sciences* (2011)
3. Bouchon-Meunier, B.: Fuzzy logic and its applications. Addison-Wesley (1995)
4. Lutz, C.: Description Logics with Concrete Domains - A Survey. In: *Advances in Modal Logics*, vol. 4. Kings College Publications (2003)
5. Eliseo, C., Di Felice, P., van Oosterom, P.: A small set of formal topological relationships suitable for end-use interaction. In: *SSD*, pp. 277–295 (1993)
6. Cullot, N., Parent, C., Spaccapietra, S., Vangenot, C.: Des SIG aux ontologies géographiques. *Revue internationale de géomatique* (2003)
7. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: *Handbook of Logic in Artificial Intelligence and Logic Programming*, pp. 439–513. Oxford University Press, Oxford (1994)
8. de Saint-Cyr, F.D., Prade, H.: Logical handling of uncertain, ontology-based, spatial information. *Fuzzy Sets and Systems, Science Direct* (2008)
9. Fisher, P.F.: Sources and consequences of error in spatial data. In: *Int Symp on the Spatial Accuracy of Natural Resource Data Bases: Unlocking the Puzzle*, pp. 8–17. ASPRS, Williamsburg, VA (1994)
10. Fonseca, F.T., Davis, C.A., Camara, G.: Bridging ontologies and conceptual schema's in geographic information integration. *GeoInformatica* 7(4), 355–378 (2003)
11. Hollunder, B.: An alternative proof method for Possibilistic Logic and Its Application to terminological logics. In: *Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 327–335. Morgan Kaufmann, San Francisco (1994)
12. Horrocks, I., Kutz, O., Sattler, U.: The Even More Irresistible SROIQ. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) *Proceedings of the 10th International Conference of Knowledge Representation and Reasoning (KR 2006)*, Lake District, UK (2006)
13. Pai, H-I.: Uncertainty management for description logic-based ontologies, Doctoral Thesis in The Department of Computer Science and Software Engineering Concordia University Montréal, Quebec, Canada (2008)
14. Pfoser, D., Tryfona, N., Jensen, C.S.: Indeterminacy and Spatiotemporal Data: Basic Definitions and Case Study. *GeoInformatica* 9(3), 211–236 (2005)

15. Qi, G., Pan, J., Ji, Q.: Possibilistic description logics of extension. In: Proceedings of the International Workshop on Description Logics (DL 2007), pp. 435–442 (2007)
16. Shu, H., Spaccapietra, S., Parent, C., Sedas, D.Q.: Uncertainty of Geographic Information and Its Support in MADS. In: ISSDQ 2003 Proceedings (2003)
17. Smith, B., Mark, D.M.: Ontology and geographic kinds (January 1, 1998)
18. Schneider, M.: Uncertainty management for spatial data in databases: Fuzzy spatial data types. In: Güting, R.H., Papadias, D., Lochovsky, F.H. (eds.) SSD 1999. LNCS, vol. 1651, pp. 330–351. Springer, Heidelberg (1999)
19. Xu, W., Huang, H.-K., Liu, X.-H.: Spatio-temporal ontology and its application in geographic information system. In: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, August 13-16 (2006)
20. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1(1), 3–28 (1978)

Ontology-Based Context-Aware Social Networks

Maha Maalej¹, Achraf Mtibaa², and Faïez Gargouri¹

MIRACL Laboratory

¹ Higher institute of computing and multimedia, University of Sfax, Tunisia

² Higher Institute of Electronics and Communication, University of Sfax, Tunisia
{maha.maalej,achrafmtibaa,faiez.gargouri}@gmail.com

Abstract. Due to the increasing progress of context-aware platforms with social networks many works use these platforms with mobile devices. Thus, we expose, in this paper, a survey of some of these works. Knowledge representation in the social networks has a great interest to obtain a set of information with a valuable signification. Therefore, we expose a state-of-the-art about the knowledge extraction using ontologies in social networks. Then, we propose, in this paper, an approach to combine these technologies (context, mobile and ontology) together to have a contextualized ontology helping to assist a mobile user in his information retrieval from the social network. We conclude by giving an idea about our future works.

Keywords: Ontology, Social network, Context, mobile device.

1 Introduction

Social networking provides progress, particularly in communication and self-expression. Hence, millions of people connect to social networks. A social network consists of people or groups connected by a set of social relationships, such as friendship, co-working or information exchange [15]. Lately, social networks have become an important mean of communication and interaction between people over the Internet. They provide many online services: email, instant messaging, file sharing, etc. Social networks are currently used in academia [26] and business communication not only in free time. A common property of Web 2.0 technologies is that they facilitate collaboration and sharing between users with low technical barriers on sites and with a limited amount of information. The basic features of a social network are profiles, friend listings, and commenting, often along with other features such as private messaging, discussion forums, blogging, and media uploading and sharing [11]. A user profile, in the social network context, is a collection of personal data associated with a specific user. A user profile can store the user's interests, gender, birthday, religious beliefs, and other characteristics of the user.

Social networking applications are changing the way of communication by using user's contextual information. As the information emerging in social networks is mainly related to the users, detecting user's context by expert becomes a crucial requirement. There are many works in literature that uses the context in social networks. Each work treats a view. Some works treat the whole notion of context as

Brézillon [7] and Wang [29]; others treat the contextual information extraction as Zitnik [34], Ghita [17], Joly [19], Narayanan [25], Damian [10] and White [30].

There is currently a significant difference between using social networking applications on a static computer compared to a mobile device, even if current mobile devices are powerful and have good connectivity. The difference is primarily related to the mobility aspect since the user contexts may change more frequently and the user may not be able to interact with the mobile device. Researchers and industry are oriented toward the use of social networks via mobile devices, given the exponential growth of mobile devices. Conversion to mobile version enables company's customers to benefit from their expertise and Smartphone instant access to the services of this company. A multitude of benefits characterizes mobile devices. Certainly, they combine practicality, ergonomics and simplicity. They are also powerful and allow easy and instant accessibility to information. Then, these mobile devices enable instant access to social networks and news. For Smartphones with Android, they provide easy access to social networking sites and good integration with Google products such as Gmail and Google Maps.

Knowledge representation represents a challenge due to the problems that it confronts. In this context, a theory which is totally based on graphs is used. In fact, graphs allow structuring concepts and relations between them. Moreover, they allow better visualization. However, the graphs do not maintain the semantics of the concepts they represent. Besides, they do not formalize the information contained in these social networks. In contrast, ontologies utilization keeps the semantic relationships between concepts with a better knowledge representation.

The rest of the paper is organized as follows. Section 2 presents major context-aware mobile platforms and applications. Then, we explore in the section 3, the semantic technologies (ontologies) and their utilization in social networks. We propose an approach, in section 4, to assist a mobile user in his information retrieval from the social network. Finally, we draw conclusion and introduce some of our future works in section 5. Section 6 presents acknowledgement.

2 Mobile Context-Aware Social Networks

Context-aware systems are computer systems that can provide relevant services and information to users by exploiting context. These systems can support rich contextual features. Such systems are available for Smartphones that are expensive devices but accepted by users in their everyday life.

The Zonezz platform [27] identifies meaningful locations such as 'home' or 'work'. It provides an easy way to understand context model and fully runs on a mobile device without the need for a central service. Other applications can use this platform to create context-awareness.

In previous work, we proposed an approach [23] to detect the requirement context by expliciting the context notion with contextual parameters. Our proposed approach is able to infer the context after knowing the contextual parameters of the user requirement. Among these proposed parameters, the parameters "Location" and the "User" which can be useful for social networks applications. We realized our proposed approach by updating the ContextOntoMR prototype [24].

Mobile devices and social networks have been widely used and are increasingly growing. Naturally, researchers aim to integrate social networks with mobile devices. The fact, that the mobile device is a personal object for the user, makes it an ideal base for the deployment of context-aware applications. Mobile devices store the social profiles of users including their interests, hobbies, etc. Some data can be extracted from the profile, with applications running on mobile devices. Profiles can also keep the dynamic context information such as location and status information.

Currently, the success of mobile applications and systems is directly linked to their potential impact on individuals and groups of people with common interests and habits [1].

Mobile Social Networks Applications enable the creation of context-aware (location-aware) applications that exploit social networks information found on existing online social networks. There are two types of context-aware mobile social networks applications: Location Aware Mobile Social networks (LAMSN) and geosocial applications. LAMSN allows exploiting social networking context with the local physical proximity of the user using mobile Smartphone like WhozThat [6], SocialAware [16] and Serendipity [12]. The geosocial applications are specialist in geo-positioning like BrightKite¹, Loopt² and Foursquare³.

After introducing major context aware social networks platforms and applications, we present in the next section some works treating mobile social networks extraction by ontologies.

3 Extracting Mobile Social Networks by Ontologies

In this section, we present different types of knowledge representation in social networks. After that, we illustrate some existing ontologies. Then, we expose some works which treated the ontology to represent the knowledge extracted from social networks.

3.1 Knowledge Representation in SN

Knowledge representation is a difficult issue due to the problems that encounter it. Social networks contain a lot of personal information about users (name, date of birth, etc.) as well as information on their friendship and their interests. Many works are required to represent this information by graphs. Mika [21] used graph theory to identify groups of users and the emergence of interest. He extended bipartite model (concepts and instances) ontologies with the social dimension (incorporating actors in the model). Then he extracted a community-based ontology from Web pages. Fan [14] proposed a framework of preserving the query graph compression, which preserves only the information needed to answer a certain query class of choice for users. Some research has used the RDF graphs to model semantic social networks. Erétéo [13] proposed a framework to exploit directly the RDF representations of social networks by using the semantic search engines on the Web. Other researchers,

¹ <http://brightkite.com/>

² www.loopt.com

³ <https://fr.foursquare.com/>

such as Corby [9] used the SPARQL query language to find paths between semantically related to RDF resources based on graphs.

Graphs allow structuring concepts and relations between them. Moreover, they allow better visualization. However, the graphs do not maintain the semantics of the concepts they represent. They do not formalize the information contained in these social networks. In contrast, ontologies utilization keeps the semantic relationships between concepts with a better knowledge representation. Indeed, ontologies are used for the specification of concepts and relationships associated with a given domain. Social networks are composed initially of entities and relationships. Thus, domain ontologies can represent these entities and relationships. Ontologies do not allow modeling conflicting information and the validity of the information encoded by reasoning. In addition, ontologies can infer new information through the inference.

3.2 Existing Ontologies

The Semantic Web is an extension of the current Web. It well defines the meaning of information, better enables computers and people to work in cooperation [2] and provides required representation mechanisms for portability between social media sites. An ontology, which is a semantic web technology, is defined by Gruber as “a shared and common understanding of a domain” [18]. Therefore, we use ontology to represent user and resource profile. The ontology-based representation is more expressive and less ambiguous. In addition, the ontology provides formal, machine-executable meaning on the concepts. Moreover, ontology standards support inference mechanisms that can be used to enhance semantic matching [20].

The FOAF (Friend Of A Friend) [3] initiative provides a way to represent social networks data in a shared and machine-readable way, since it defines an ontology for representing people and the relationships that they share. While the SIOC (Semantically Interlinked Online Community) [4] project was initially established to describe and link discussion posts taking place on online community forums such as blogs, message boards, and mailing lists. By using agreed-upon semantic Web formats like FOAF and SIOC to describe people, content objects, and their connections, social media sites can interoperate and provide portable data by appealing to some common semantics [5]. As discussions begin to move beyond simple text-based conversations to include audio and video content, SIOC has evolved to describe not only conventional discussion platforms but also new Web-based communication and content-sharing mechanisms.

3.3 Extracting Mobile Social Networks by Semantic Technologies

There are many works that treat knowledge extraction from social networks using ontology. The treated works are divided into two categories knowledge extraction from profiles and knowledge extraction from tags. The first category is represented in table 1. Table 2 presents the second category. The comparison characteristics are ontology source, purpose of creating the ontology, social network(s) used for the extraction and the used extraction tool.

In the table 1, we present the works that extract knowledge from user profiles (Whitsitt [31], Challenger [8], and Yadav [32]). We notice that the social network the

most used is Facebook. The source of the used ontology for the knowledge extraction differs: from scratch or from an existing ontology.

Table 1. Our comparison of knowledge extraction: from user profiles

	Ontology source	Purpose of creating the ontology	Social network used for the extraction	Used extraction tool
Whitsitt [31]	From a graph	To infer implicit relationships and selection of relevant links	Facebook	ATRAP
Challenger [8]	From FOAF, SIOC, DBLP ontologies	To integrate different data sources (academic social network: education of a person)	LinkedIn , Facebook	Social Graph API
Yadav [32]	From micro posts shared in SN	To specify signification of the semantic annotations of Web resources	Facebook	Concept Extractor + VIBHAKTI PARSER

Table 2 compares the works of (Ying [33], Monachesi [22] and Veres [28]). We perceive that the source of ontology differs: from only tagged data sources or from tagged data sources combined with an existing ontology. We observe here that the social network Delicious is the most used in these works.

Table 2. Our comparison of knowledge extraction: from tags

	Ontology source	Purpose of creating the ontology	Social network used for the extraction	Used extraction tool
Ying [33]	From tagging activities	To model tagging data and keep the semantics of metadata	Delicious, Flickr, Youtube	Smart and Simple Webcrawler framework
Monachesi [22]	LT4eL domain ontology + Dbpedia + MOAT ontology + SKOS vocabulary + social tagging data	To support informal learning	Delicious, YouTube and Slideshare	A crawler that uses APIs provided by the social networking applications

Veres [28]	From any tagged data source	To organize bookmarks and other information sources	bookmarks that had already been semantically annotated with WordNet synsets	Not available
-------------------	-----------------------------	---	---	---------------

4 Ontology-Based Context-Aware Mobile Social Networks

We propose an approach in order to assist the mobile user in his search on social network. This approach is composed of five steps as presented in figure 1.

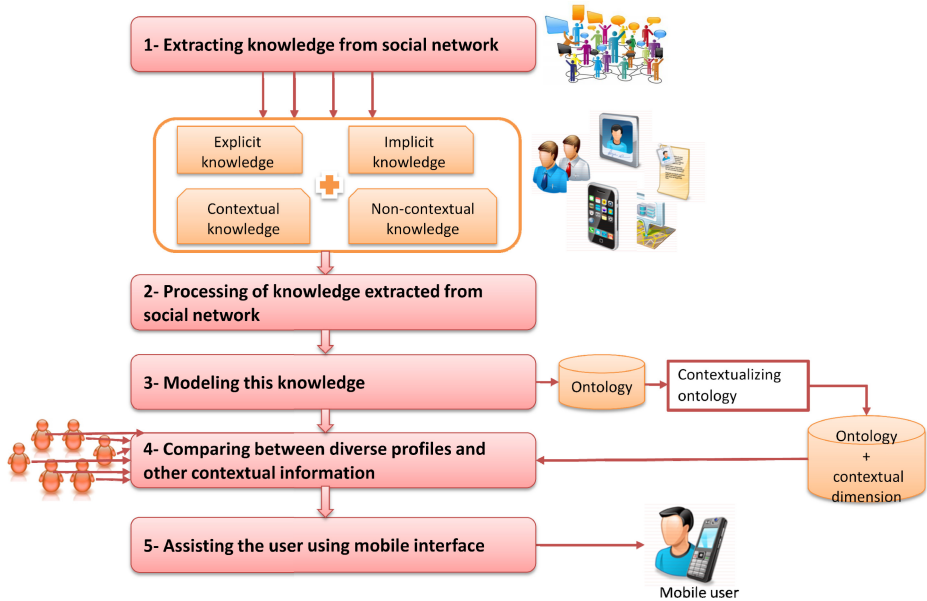


Fig. 1. Our proposed approach to build an ontology based context aware mobile social network

Extracting knowledge from social network

This step allows extracting diverse form of knowledge explicit, implicit, contextual and non contextual, from social network. This knowledge represents the raw data extracted by the tool or the API. Then, we keep the useful knowledge for our ontology by the next step.

Processing of knowledge extracted from social network

This step permits to process the extracted amount of knowledge which contains additional information which is not useful for our work. Thus, we maintain only the required data.

Modeling the processed knowledge

By modeling the knowledge, we mean that we use the knowledge about users and the contextual knowledge to create our ontology. The user's general information, their interests, their shared data their comments, etc, represent the effective knowledge. We build an ontology from this processed data to profit from the advantages of sharing knowledge and keeping semantics of ontology construction. The ontology will contain the main concepts of social network, their properties, their relationships and some axioms controlling the structure of the ontology. By contextual information we mean the information related to the user when he is searching something from the SN. This contextual information is represented as a contextual dimension containing the parameters needed to ameliorate the user's search.

Comparing between diverse profiles and others contextual information

This step consists of comparing the different user profiles and the contextual information of mobile user. This comparison is achieved by using tools and algorithms to choose the best result of the user query

Assisting the user using mobile interface

The last step permits assisting the user, by mobile interface. The result should suit the mobile performance (size of the screen, size of memory, etc).

5 Conclusion and Future Works

During this paper, we exposed the major context-aware platforms and applications related to social networks. As the ontology is an essential element in the semantic web technologies and its big role in managing knowledge, we assume that is necessary to make a tour over the existing works that use ontology to extract knowledge from the social networks. Indeed, we presented our classification of some works treating knowledge extraction from social networks using ontology. Then we introduced our proposed approach to assist the mobile user in his information retrieval from the social network. In our future works, we intend to contextualize our ontology through a method that extracts information from a social network using ontology.

Acknowledgements. This work is supported by the Tunisian-Algerian program of co-operation in science & technology: Towards a new Manner to use Affordable Technologies and Social Networks to Improve Business for Women in Emerging Countries.

References

1. Arnaboldi, V., Conti, M., Delmastro, F.: CAMEO: a novel Context-Aware MiddleWare for Opportunistic Mobile Social Networks. IIT TR-02/2012 Technical report. Institute of Informatics and Telematics (March 2012)

2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* 284(5), 34–44 (2001)
3. Brickley, D., Miller, L.: FOAF Vocabulary Specification. Namespace Document, FOAF Project (September 2, 2004), <http://xmlns.com/foaf/0.1/>
4. Breslin, J.G., Harth, A., Bojars, U., Decker, S.: Towards Semantically-Interlinked Online Communities. In: Gómez-Pérez, A., Euzenat, J. (eds.) *ESWC 2005*. LNCS, vol. 3532, pp. 500–514. Springer, Heidelberg (2005)
5. Bojars, U., Breslin, J., Finn, A., Decker, S.: Using the Semantic Web for Linking and Reusing Data Across Web 2.0 Communities. *The Journal of Web Semantics, Special Issue on the Semantic Web and Web 2.0* (2008)
6. Beach, G.M., Akkala, S., Elston, J., Kelley, J., Nishimoto, K., Ray, B., Razgulin, S., Sundaresan, K., Surendar, B., Terada, M., Han, R.: Whozthat? Evolving an ecosystem for context-aware mobile social networks. *IEEE Network* 22(4), 50–55 (2008)
7. Brézillon, P., Marie Curie, P.: A context approach of social networks. In: *Proceedings of the Workshop on Modeling and Retrieval of Context* (2004)
8. Challenger, M.: The Ontology and Architecture for an Academic Social Network. *IJCSI International Journal of Computer Science Issues* 9(2(1)) (March 2012) ISSN (Online): 1694-0814
9. Corby, O.: Web, Graphs and Semantics. In: Eklund, P., Haemmerlé, O. (eds.) *ICCS 2008*. LNCS (LNAI), vol. 5113, pp. 43–61. Springer, Heidelberg (2008)
10. Damian, A., Nejdil, W., Paiu, R.: Peer-Sensitive ObjectRank-Valuing Contextual Information in Social Networks. In: *Proc. of the International Conference on Web Information Systems Engineering* (2005)
11. Domingue, J., Fensel, D., Hendler, J.A. (eds.): *Handbook of Semantic Web Technologies*. Springer, Heidelberg (2011), doi:10.1007/978-3-540-92913-0
12. Eagle, N., Pentland, A.: Social serendipity: Mobilizing social software. *IEEE Pervasive Computing* 4(2), 28–34 (2005)
13. Erétéo, G., Gandon, F., Corby, O., Buffa, M.: *Semantic Social Network Analysis*. Web Science (2009)
14. Fan, W.: Graph pattern matching revised for social network analysis. In: *Proceedings of the 15th International Conference on Database Theory, New York USA*, pp. 8–21 (2012) ISBN: 978-1-4503-0791-8
15. Garton, L., Haythornthwaite, C., Haythornthwaite, C.: Studying online social networks. *Journal of Computer-Mediated Communication* 3 (1997)
16. Gartrell, C.M.: *Socialaware: Context-aware multimedia presentation via mobile social networks*. Master's thesis, University of Colorado at Boulder (December 2008)
17. Ghita, S., Nejdil, W., Paiu, R.: Semantically Rich Recommendations in Social Networks for Sharing, Exchanging and Ranking Semantic Context. In: *ESWC Workshop on Ontologies in P2P Communities* (2005)
18. Gruber, W.A., Boudreaux, J.C.: *Intelligent manufacturing: programming environments for CIM*. Springer, London (1993)
19. Joly, A., Maret, P., Daigremont, J.: Context-Awareness, the Missing Block of Social Networking. *International Journal of Computer Science and Applications* 4(2), 50–65 (2009) ISSN 0972-9038
20. Li, J., Wang, H., Khan, S.U.: A Semantics-based Approach to Large-Scale Mobile Social Networking. *Mobile Networks and Applications* 17(2), 192–205 (2012)
21. Mika, P.: Ontologies are us: a unified Model of Social Networks and Semantics. In: *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5(1), pp. 5–15. Elsevier Science Publishers B. V., Amsterdam (2007)

22. Monachesi, P., Markus, T.: Using Social Media for Ontology Enrichment. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) *ESWC 2010, Part II. LNCS*, vol. 6089, pp. 166–180. Springer, Heidelberg (2010)
23. Mtibaa, A., Maalej, M., Gargouri, F.: Context detection in ontology. *IADIS International Journal on Computer Science and Information System (IJCSIS)* 1.7(2), 117–128 (2012) ISBN: ISSN: 1646-3692
24. Mtibaa, A.: Une ontologie de multi-représentation pour la spécification des besoins multi-contextes. Ph.D. thesis, MIRACL Laboratory, University of Sfax, Tunisia (2011)
25. Narayanan, A., Shmatikov, V.: De-anonymizing Social Networks. In: *IEEE Symposium on Security and Privacy*, pp. 173–187 (2009)
26. Pemppek, T.A., Yermolayeva, Y.A., Calvert, S.L.: College students' social networking experiences on Facebook. *Journal of Applied Developmental Psychology* 30(3), 227–238 (2009), <http://dx.doi.org/10.1016/j.appdev.2008.12.010> (retrieved)
27. Roth, J.: Context-aware apps with the Zonezz platform. In: *MobiHeld 2011 Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds*, Article No. 10 (2011)
28. Veres, C., Chen, W., Opdahl, A., Johansen, K.: Ontology Extraction from Social Semantic Tags. In: *International Conference on Web Intelligence, Mining and Semantics*, Sogndal (May 2011)
29. Wang, X., et al.: Ontology based context modelling and reasoning using owl. In: *Workshop on Context Modeling and Reasoning at IEEE Int'l Conference on Pervasive Computing and Communication, USA* (2004)
30. White, R.W., Bailey, P., Chen, L.: Predicting User Interests from Contextual Information. In: *SIGIR 2009 Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, USA, pp. 363–370 (2009)
31. Whitsitt, S.G.A., Sangman, C., Sprinkle, J., Ramasubramanian, S., Suantak, L., Rozenblit, J.: On the Extraction and Analysis of a Social Network with Partial Organizational Observation. In: *IEEE 19th International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, April 11-13, pp. 249–256 (2012)
32. Yadav, B.: Microposts' ontology construction via concept extraction. *International Journal of Web & Semantic Technology* (2012) pubid: 103-505-145
33. Ding, Y., Jacob, E.K., Fried, M., Toma, I., Yan, E., Foo, S., Milojević, S.: Upper tag ontology for integrating social tagging data. *Journal of the American Society for Information Science and Technology*, Vol 61(3), 505–521 (2010)
34. Zitnik, S.: Collective Ontology-based Information. In: *CAiSE 2012*, Gdańsk, Poland, June 25-29 (2012)

Diversity in a Semantic Recommender System

Latifa Baba-Hamed and Magloire Namber

RIIR Laboratory, Computer Science Department, University of Oran, Algeria
lbabahamed@yahoo.fr

Abstract. In this paper, we introduce the notion of diversity in the recommender systems (RS). The aim is to provide the user with not only all the most relevant contents, but also the most diversified. To do this, we have developed a diversification algorithm that we have implemented on a semantic RS. This last performs the matching between the description of the contents and the user profile. A comparison of our algorithm to the diversity algorithm Swap, in terms of relevance and diversity, has revealed better results.

Keywords: user profile, preferences, recommendation system, relevance, diversity.

1 Introduction

With the development of Web 2.0, the proliferation of community web sites (eg Blogs, Forums) and the generalization of the use of social networks (eg Facebook, Tweeter, LinkedIn), access to relevant information has become a real challenge for users. Indeed, the overabundance of information has led to the deterioration of the quality of results returned by the web to users. Recommender systems (RS) are effective tools to overcome the problem of information overload by providing users with relevant contents.

Recommendation strategies are based generally on collaborative filtering (CF), content-based filtering (CBF) or a combination of these two approaches (hybrid systems) [4], [10]. The CF systems recommend products based on the similarity of the preferences of a group of customers known as neighbors. This assumes that users with common interests in the past will continue probably to share the same interests in the future [7], [9]. CBF systems use matching between a user profile and content descriptors to recommend appropriate products [8], [11].

Conventional filtering systems suffer from a latent problem that manifests itself by recommending a set of highly similar contents. For example, in the case of movie recommendation, this problem is reflected in the recommendation of a set of films of the same genre (Action, Drama, etc.), which may annoy the active user. Therefore, the concepts of diversity and novelty have been introduced by researchers to address this problem.

The goal of diversification recommendations is to identify a list of items that are dissimilar with each other, but nonetheless relevant to the users interests. Thus, a recommender system is more effective if it achieves a good balance between relevance and diversity.

The research works in the area of diversity-aware RS can be divided into four categories: the attribute-based approaches, the explanation-based approaches, the approaches based on the model of communities' spaces and approaches using ranking-based techniques. In the first category the diversity of the recommended list is defined by how much each item in the list differs from the others in terms of their attribute values [2]. In the second category, for example, for content-based strategies, the explanation is defined for a recommended item as a set of similar items that the user has highly rated in the past. The premise for explanation-based diversification in this case is that for two different recommended items i and j , the closer their explanations, the more homogeneous i and j [5], [6]. The third category proposes the principle of formation of communities based on several criteria. For example, in a movie recommendation system, we can associate the criteria *professions* and *city*, to the *evaluation* criterion for the formation of communities. A user U with the *researcher profession*, living in the *city of Oran* and having evaluated a number of films, will be assigned to three different communities: the community relating to the evaluations (whose members are the users whose evaluations are similar to those of U), the community relating to the city of residence, and the community relating to the researcher profession. The user can thus benefit from the recommendation lists resulting of the three communities. This could help to diversify his contents [9]. The fourth category is based on the fact that to recommend the most popular items increases relevance. Intuitively, recommend less popular items saves diversity. This approach is based on the popularity to recommend products with diversified contents. One way to approach this is to keep products with a relevance value greater than a predefined threshold, then to order them in descending order of their popularity [1]. Another category of works takes into account the personality traits of a user to recommend different products/services [15].

In this paper, we present a diversity-aware RS which uses the CBF filtering strategy and allows the simultaneous consideration of the semantic aspect of a user's profile and its quantitative preferences. To do this, we have developed a diversity algorithm called DivUse which we have implemented on a semantic RS. The semantic aspect of the RS is reflected by using a similarity measure which combines the semantic aspect of the user profile and the quantitative preferences. Taking into account the semantic aspect is mainly done by measuring the similarity between sets of concepts belonging to an ontology (for example an ontology of films). Then, we compared our algorithm to the diversity algorithm Swap [6] in terms of diversity and relevance. We are based in this comparison on the test platform made during the project APMD [12], which consists of the integration of the two databases of films, namely IMDB [13] that provides the characteristics of the films (title, genre, actors, director, etc.) and MovieLens [14] that provides the ratings given by users for certain movies. These scores are between 1 and 5.

The rest of the paper is organized as follows. In section 2, we introduce some basic concepts related to recommender systems. Section 3 is devoted to the architecture of our diversity-aware system and the elaborated algorithm of diversity.

In Section 4, we present the results of the comparison of the developed algorithm DivUse and the algorithm *Swap* in terms of diversity and pertinence depending on the CBF strategy. Finally, Section 6 concludes the paper and gives some perspectives.

2 Basic Concepts

This section covers some fundamental concepts in recommender systems. This includes user profiles, user preferences, and matching operators.

Profile. It can be thought of as a summary of previous user activity. It encompasses user's needs and preferences. In other words, it is a user model and a source of knowledge acquisition that contains all aspects related to the user.

Preference. Preferences are an integral part of user profiles. A preference is a formula for prioritizing a set of objects in relation to the interests and needs of a user. There are several kinds of preferences. A presentation of all these types is given in [16]. In this article, we focus on quantitative preferences which are expressed through scoring functions which assign scores to different objects. Such preferences are used to define a total order on objects. They are modeled as follows: a preference pi is a pair (pri, wi) where pri is a predicate of type $\langle \text{attribute, operator, value} \rangle$ and wi is a real number between 0 and 1 which represents the degree of interest of the user with respect to the predicate pri . 0 reflects the minimal preference and 1 reflects the maximum preference.

Matching Operator. At the heart of most recommender systems, we find a matching operator that measures the similarity between two user profiles, two content descriptors or the similarity between a user profile and a content descriptor. Such as user profiles and content descriptors are often modeled with vectors of weighted keywords, only the vector measurements as Cosine and Pearson correlation were used. However, the advent of the Semantic Web and ontology development have placed at our disposal a wide range of semantic similarity measures that can complement the vector measurements mentioned above. A classification of these measures is given in [3].

3 General Architecture of the System

As we mentioned above, we propose to recommend diversified items to a given user by application of the CBF filtering technique. To this end, we developed and implemented an algorithm of diversity (DivUse) which we integrated in the semantic CBF system. Then, we compared DivUse with Swap algorithm in terms

of relevance and diversity. We dedicate this section to detail the components of this system. For illustrative purposes, we chose the cinematographic field (i.e., we opted for films like contents), but the same procedure remains valid for other types of contents such as: books, research articles, restaurants, songs, web pages, etc.

The architecture of our system consists of several modules as shown in Figure 1. Its different modules are presented in the following.

3.1 User Profile Builder

Without loss of generality, we define the user profile as a set of preferences $P_u = \{p_1 \dots p_n\}$. For example, let R be a relation schema modeling cars, R (type, price, color, mileage, power, nbPlaces). A user can set the following preferences: p_1 ($\langle \text{color}, '=', \text{'Red'} \rangle, 0.9$) and p_2 ($\langle \text{Price}, '>', 1.4\text{M-DA} \rangle, 0.2$).

In our system, the user profile can be explicit or implicit. We talk about explicit profile when it is the user who provides his preferences through our acquisition platform. This is done by assigning weights of importance to different film genres recognized by our system. When the profile is directly calculated from the consumptions (feedback) of a user, then we say that it is implicit. Implicit quantitative preferences are then calculated as follows: If a user u consumes two films of *Action* genre with respective scores of 5 and 4, then the user's preference for the genre *Action* is equal to 4.5 ($= (5 + 4) / 2$). A normalization of the user preferences is needed before using them for the calculation of the prediction. This normalization is to set preferences between 1 and 5, in an interval $[0, 1]$.

In order to apply a numeric similarity measure on our user profile, we defined a vector representation of the profiles as follows: let P is the set of preferences of a profile and Op an arbitrary but fixed order on the preferences that belong to P . We refer to the i th element of P according to the order Op by $P[i]$. The vector representation of the profile P is a numeric vector (of real type) V of dimension N in which the i th element of V is the weight of the preference $P[i]$ [3].

3.2 Content Descriptor Builder

A product is characterized by several properties. Some of these properties constitute the product descriptor. In our case, the product which is a film is characterized by the properties (title, director, actors, genre, etc.). The descriptor that we have chosen is limited to the concepts genre and title. It is represented by a vector whose columns are the genres of the films and each cell can take the value 1 if the film has the genre expressed in column, otherwise it takes the value 0. The genres that we have used are: *Adventure, Action, Fantasy, Sci-Fi, War, Western, Biography, History, Drama, Comedy, Musical, Romance, Family, Animation, Sports, Thriller, Crime, Horror, and Mystery*.

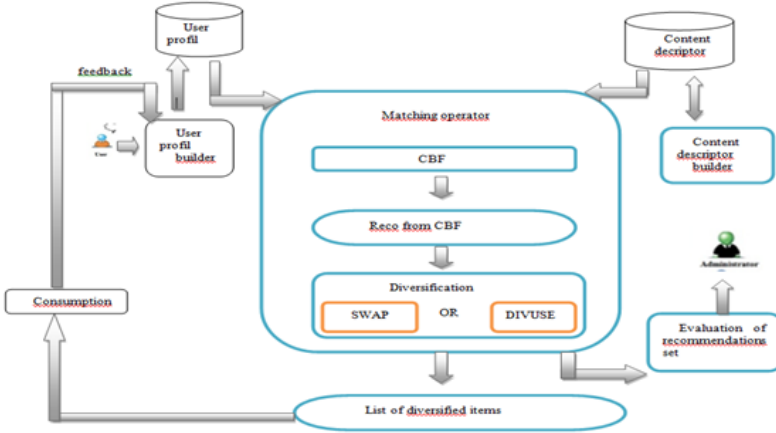


Fig. 1. General architecture of the system

3.3 Matching Operator

This module constitutes the most important part of our system. It includes two sub-modules : the sub-module that provides recommendations based on the technique of CBF and the sub-module that takes into account the diversity of the recommendations. The first module uses a hybrid measure of semantic similarity $Sim_{globale}$ which is presented in [3]. This measure combines the advantages of semantic similarity metrics (matching of profiles and films) and those of numeric similarity metrics (matching of preferences).

We have combined the semantic similarity measure of Contrath & Jiang [17] with the numeric similarity measure of Pearson to improve and increase the relevance of responses to the user. $Sim_{globale}$ includes three levels simultaneously: the position of the concepts in the ontology hierarchy (position of the movie genres in the movies ontology), the information content of concepts (probability of occurrence of a movie genre), and the weight of user’s preferences. We recall the formula for this measure below:

$$Sim_{globale}(Pu, C) = \alpha \times Sim_{sem}(Pu, C) + \beta \times Sim_{pref}(\vec{P_u}, \vec{C}). \quad (1)$$

With the sum of the two coefficients equals to 1 ($\alpha + \beta = 1$). Sim_{sem} represents the value of the semantic similarity between the profile and the content by using the measure of Jiang & Contrath. Sim_{pref} is the similarity value obtained by applying the Pearson correlation to vector representations of the profile and the content (the movie). This module uses the similarity $Sim_{globale}$ to perform the matching between the user profile and the descriptors of products. It provides a list of recommendations ranked according to the decreasing relevance.

The second sub-module ensures the diversification of the recommendation list obtained by the first sub-module according to our diversity algorithm DivUse or according to the diversity algorithm Swap given in [6]. We present in what follows, the details of *DivUse* and the principle of *Swap*.

Algorithm DivUse. DivUse is based on a principle of utility. We assume that each item has a degree of usefulness in relation to the user. This utility is based on the relevance and the novelty of the item. Novelty expresses the fact that no similar item has been recommended to the user. The novelty of an item decreases, if it is similar to items already consumed by the active user. Initially all items have a novelty equal to 100%. DivUse selects the most relevant item in the list of recommendations produced by the CBF sub-module. Then it updates the novelty of all similar items to that which has been selected. The utility of an item F for an active user Ua is calculated using the formula (2). The relevance (pertinence) of an item F represents the correlation of its descriptor to the profile of the active user Ua . This correlation is given by the formula of Pearson. The novelty is the complement of the redundancy.

Input: $L1, K$ (initial List and number of recommendations)
Output: $L1$ (the result list initially empty)

- 1 – $L1 = L1.sortedByRelevance()/*$ decreasing order;
- 2 – $L2.add(firstiteminL1)$;
- 3 – $L1.remove(firstitem)$;

Repeat ;

for *eachitemiinL1* **do**

a. Redondancy(i)= $FU * \frac{\Sigma sim(i,i')}{lengthofL2} / * i'$ item in L2 ;
b. Novelty(i)=1- Redondancy(i);
c. Utility(i)= <i>similarity(i) * novelty(i)</i> ;

end

- $L2.add(i) /*i$ in L1 with highest utility ;
- $L1.remove(i)$;
- 6- Until $L2.size()=K$;

return $L2$

Algorithm 1. Algorithm DivUse

The novelty of F , knowing that the article G has already been consumed by the user Ua is calculated using the formula (3). The novelty of F , knowing that the set of items $f1$ fn has already been consumed by the user Ua is calculated using the formula (4). Once the calculation of the utility is given, the item having the highest utility value is placed in the result list. The process is repeated until reaching the desired number of items (e.g. the top 10 items). FU represents the diversity parameter. It is used to estimate the redundancy of a product F for a given user U . $FU = 1$ implies that items similar to those which the user has already consumed are redundant with a percentage of 100%. $FU = 0.4$ implies that products similar to those which the user has already consumed are redundant with a percentage of 40%. The items whose redundancy is equal to 0 will have a novelty equal to 1 (i.e. they are new with a percentage of 100%).

$$Utility(Ua, F) = Pertinence(Ua, F) * Novelty(Ua, F) \quad (2)$$

$$Novelty(Ua, F/G) = 1 - FU * Similarity(F, G) \quad (3)$$

$$Novelty(Ua, F/\{f1, fn\}) = 1 - FU * (\sum_{i=1}^n Similarity(F/f_i))/n \quad (4)$$

Algorithm Swap. Swap is based on the principle of the permutation of the items. From a set S of candidate items which are obtained by CBF or CF, Swap selects the top K products (K most relevant products of S) and puts them in a list L. From this list L, the algorithm retains the first product and calculates its average distance with respect to the k-1 other products. It repeats this process for the remaining k-1 items. Then it considers among the k items in the list L, the product $P_{dist-min}$ whose average distance is lowest. Then it considers the k +1th element of the list S to compare its relevance to the Product $P_{dist-min}$. To make the comparison, it performs the difference of relevance between $P_{dist-min}$ and k +1th element of the list S. If this difference is less than the threshold of relevance (UB), then the algorithm computes the average distance of the k +1th element with respect to the list L. If this average of distance is greater than that of the item $P_{dist-min}$, then Pdist-min is swapped with the k +1th element of the list S. The process is repeated until the difference of pertinence between the product to swap ($P_{dist-min}$) and that of the k +1th product is greater than the threshold UB or there are no products in S.

3.4 Consumption

This module allows the user to consume and to evaluate the list of recommendations which is provided to him. The active user can rate items on a scale from 1 to 5. These evaluations will allow updating his profile in order to refine the results of his research in the coming sessions.

3.5 Evaluation

This module is responsible for the evaluation of our prototype in terms of diversity and relevance.

4 Evaluation and Comparison

This section is dedicated to the evaluation of our algorithm of diversity DivUse. To do this, we used two metrics: diversity and relevance. Indeed, the objective here is to create a balance between diversity and relevance of recommended products. Diversity allows us to estimate the average distances between recommended products.

We describe, in a first step, the databases used. Then, we present and interpret the curves of relevance and diversity obtained by our algorithm using the CBF technique. Finally, we give a comparison between the two algorithms DivUse

(our algorithm) and Swap (the diversity algorithm given in [6]. To perform our experiments and demonstrate the feasibility of our application, we consider the cinematographic field.

The data used for the evaluation of our algorithms are based on the platform of test performed during the project APMD (or PAMD: Personalized Access to Masses of Data), which integrates two databases on films, namely, IMDB (Internet Movie Database) which provides the film characteristics (title, genre, actors, director, etc.) and MovieLens which provides the ratings given by users for certain movies. These scores are between 1 and 5. This evaluation required the tables:

- User-ratings: in which one million one hundred ninety four (1000194) evaluations are listed.
- I-movies: This contains 3704 movies.
- Users: in where are recorded six thousand and forty (6040) users.
- Im-oviegenres: This contains the identifiers and the genres of films.

Experimentation has focused on evaluating the mean of relevance and diversity of recommendation lists generated for a user. In this module, we have considered the users profile chosen at random among those of the system users. We have recommended movies by executing the algorithms DivUse and Swap, and this is by varying the diversity parameter FU (case of DivUse) or UB (case of Swap). The diversity parameters are initially set to 0 (no diversity recommendations). We have varied them at intervals of 0.2 (which has allowed us to plot the curves of diversity and relevance). Then, for each fixed parameter value, we have calculated the average diversity and the average relevance of the generated list of recommendations. The idea is to identify the algorithm which makes a balance between diversity and relevance. This algorithm must have a good increase of relevance with a slight decrease in diversity.

We present in what follows, the different graphs and interpretations of our experiments in the case of the CBF technique.

We can see in Figure 2-b (which concerns the algorithm Swap with CBF strategy) that the diversity always prevails over relevance (we can see that for $UB = 0.2$, we have a pertinence average = 0.5 while the diversity = 0.65: relevant and diversified). We have, therefore, a slight drop of relevance vs. a rapid growth of diversity.

In Figure 2-a, we note that with the algorithm DivUse by the CBF approach, we gain diversity without losing relevance. We can see that DivUse increases the diversity from 0.47 to 0.72 for a value of FU varying from 0 to 0.4. Diversity is stationary beyond $FU = 0.4$. DivUse slightly lowers the relevance from 0.54 to 0.52 for FU varying from 0 to 0.7.

By comparing Figure 2-a and Figure 2-b (representing DivUse and Swap respectively), it appears that the algorithm DivUse gives better results than the algorithm Swap because it does not lose relevance at the expense of diversity.

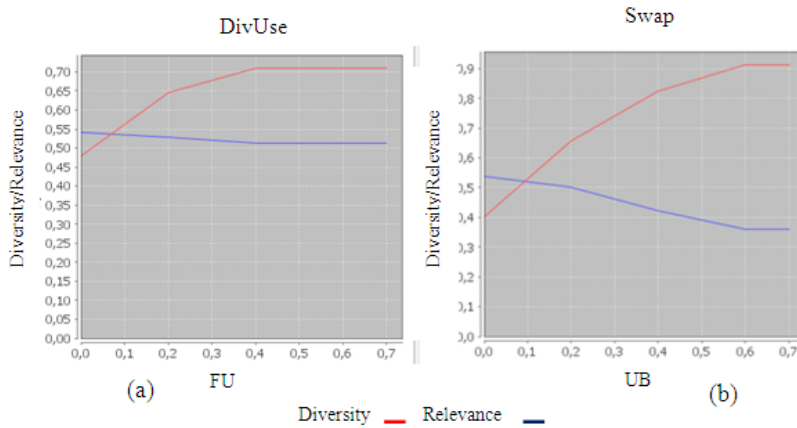


Fig. 2. Diversity and relevance according to DivUse and Swap algorithms using the CBF technique

5 Conclusion

In this work, we were interested in the implementing a recommendation system, taking into account the notion of diversity. To do this we have developed an algorithm of diversity DivUse which we compared to the algorithm Swap. The results of this comparison showed that DivUse is better than Swap since it provides a better balance between diversity and relevance. We plan to improve the performance of our prototype by considering the optimization of the algorithm DivUse in terms of execution time using indexes to improve access to the database.

References

1. Adomavicius, G., Ok Kwon, Y.: Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Data Eng.*, 1041–4347 (2011)
2. Angel, A., Koudas, N.: Efficient Diversity-Aware Search, University Toronto. In: *Proceedings of the 2011 International Conference on Management of Data (SIGMOD 2011)*, pp. 781–792. ACM, New York (2011)
3. Baba-Hamed, L., Abbar, S., Soltani, R., Bouzeghoub, M.: Elaboration et Evaluation d'un Système de Recommandation Sémantique. In: *Proceedings 1st International Conference on Information Systems and Technologies*, April 24–26, pp. 515–523 (2011)
4. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. *Journal of Personalization Research, User Modeling and User-Adapted Interaction* 4, 331–370 (2002)
5. Yu, C., Lakshmanan, V.S., Amer-Yahia, S.: Recommendation Diversification Using Explanations. In: *ICDE*, pp. 1299–1302 (2009)

6. Yu, C., Lakshmanan, V.S., Amer-Yahia, S.: It takes variety to make a world: diversification in recommender systems. In: EDBT, pp. 368–378 (2009)
7. Herlocker, J.L., Konstan, J.A., Terven, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. *ACM Transaction on Information Systems* (2004)
8. Markov, K., Ivanova, K.: An ontology-content-based filtering method. In: Proceedings of the Fifth International Conference on Information Research and Applications, Varna, Bulgaria (June 2007)
9. Nguyen: COCoFil2: Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés. Thèse de Docteur. Université Joseph Fourier, Grenoble I (Novembre 2006)
10. Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. Technical report, University of California, Irvine (1999)
11. Shoval, P., Maidel, V., Shapira, B.: An Ontology- Content-Based Filtering Method. In: I.Tech-2007, Information Research and Applications (2007)
12. APMD, <http://apmd.prism.uvsq.fr>
13. IMDB (Internet Movie Database), <http://www.imdb.com>
14. MovieLens, <http://movielens.umn.edu>
15. Vallet, D., Castells, P.: Personalized Diversification of Search Results. In: 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012), Portland, OR, USA (August 2012)
16. Ouardas, F.L., Baba-Hamed, L., Abbar, S.: A Survey on Personalization Approaches Using Positive/Negative Preferences. In: Proceedings of 3rd International Conference on Information Systems and Technologies (ICIST 2013), Tangier, Morocco, March 22- 24 (2013)
17. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of the 10th International Conference on Research in Computational Linguistics, Taiwan (1998)

Ontology - Driven Observer Pattern

Amrita Chaturvedi and Prabhakar T.V.

Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur, Kanpur-208 016, India
{amrita,tvp}@iitk.ac.in

Abstract. We propose an ontology driven observer pattern which not only mitigates the drawbacks identified in the GoF observer pattern but also mitigates the drawbacks which occur in the general usage of patterns. We separate and encapsulate the pattern logic in an ontology component which increases the reusability of the pattern at the implementation level as well. The proposed solution enables to change the classes participating in the pattern even at runtime. Even the users/non-programmers can make changes in the pattern to change the application behavior. It enables identification of a pattern present in a code and also allows easy change, addition/removal of the pattern to/from the code. The proposed pattern also decouples the participant classes from each other thereby enhancing the reusability and modifiability of each of the participant classes.

Keywords: Observer pattern, ontology, maintainability, reusability.

1 Introduction

Observer pattern is classified as behavioral pattern in GoF patterns catalogue [1]. Its main design intent is to define multiple dependency relationship between objects so that when an object (observable or subject) changes state, all its dependents (observers) are notified and updated automatically. Although observer pattern is described [1] to enhance reusability of the components, there are certain drawbacks which have been identified to exist in observer pattern. The description of drawbacks identified in the observer pattern is given in section 3.1.

Consider an example of an interactive application like a general demographics survey information system. Typically, different screens display the same information but in different formats. For example a screen S_1 displays median income of a population for a set of 20 years of both the genders and all age groups in a tabular format. Other screens may display the same information in pie chart, bar chart, line chart etc. There may also be sub views of a particular screen. For example, S_1 may have a sub view S_{11} which displays median income of all age groups for a set of 20 years of a particular gender say males. Another sub view S_{12} displays median income of a particular age group (say 18-29) for a set of 20 years of both the genders. Now, suppose it is a system requirement that tabular formats of all the main screens (example S_1) should be updated prior to the other screens. It is difficult to accommodate this requirement in the code of the

observer pattern and even more difficult to change the update priorities. Also, suppose the median income of female population of the age group of 30-49 for a set of some years is added in the model (subject) which needs to be reflected in all the screens observing the model. In this case only the main screen S_1 needs to be updated and not S_{11} and S_{12} (because they don't display the information that has changed). However, in traditional observer pattern it is very difficult to specify the specific attributes of the subject on which the observers depend. So all the observers completely or partially depending upon the subject are notified of changes and are updated even if they don't depend on the information that has changed.

It is also not possible to make runtime changes in the classes playing the subject and observer roles. Suppose instead of the median income being displayed by the screens in different formats, it is desired to display the per capita income of the population for the same set of years. Then, to accommodate these changes some code requires to be changed and the whole application needs to be recompiled. User cannot make these changes in runtime environment. However, if this design pattern logic were extracted out in a separate component and the application specific details were encoded declaratively (as in our observer ontology) then the user could be allowed to make runtime changes in the application level details of the design pattern. This can be done by providing a GUI for the ontology where the user can change the behavior of the application by tweaking the observation rules in the ontology through its GUI. Also the finer observation rules about which observers depend upon which particular attributes of the subject can also be encoded in the ontology thus removing unnecessary notifications and updates.

Apart from the drawbacks specific to the observer pattern, there are certain drawbacks which are present in design patterns usage in general. These are discussed in section 3.2.

We propose an ontology driven observer pattern to mitigate all the above problems. We have built a generic observer ontology, details of which are given in section 4.1. Ontology is one of the most widely used semantic technologies and is formally de-fined as "an explicit specification of a shared conceptualization"[2]. It can be used to model domain knowledge in the form of a set of concepts and relationships between them and can be expressed in ontology languages like OWL, OIL, DAML etc. It is mostly used for declarative knowledge maintenance and handling. Our observer ontology contains the design pattern logic of the observer pattern at the intension level (schema). The application specific details are encoded at the extension level (instances of the ontology concepts). Whenever a subject property (attribute) changes all the observers depending upon that property are retrieved from the ontology and are updated by the ontology manager.

The paper is structured as follows: section 2 covers the related research on observer pattern in specific and design patterns in general. It details the different techniques which various researchers have proposed to overcome the problems of patterns and how our proposed technique is better and/or different from the existing ones. Section 3 highlights the problems identified in the observer

pattern and the pattern usage in general. Section 4 describes the static, dynamic and implementation aspects of the proposed ontology driven observer pattern. Section 5 summarizes the conclusion and ultimately the references are enlisted.

2 Related Research

Hachani and Bardou [3] present a motivating work to start using aspect oriented programming (AOP) to implement object oriented design patterns. They separate the code related to the application of pattern from the rest of the code and encapsulate it in aspects. Hannemann and Kiczales [4] provide a similar research and present a comparison of the AspectJ and Java implementations of concrete instances of the GoF design patterns. They identify that in the structure of the observer pattern, some parts are common to all implementations of the pattern, and other parts are specific to its particular application. They present an AspectJ code that reflects this separation of reusable and application-specific parts. Garcia *et al* [5] propose a quantitative assessment of java and AspectJ implementations of GoF design patterns which complements the work by Hannemann and Kiczales [4]. One problem with these approaches is that, runtime changes in the pattern participants and their dependencies cannot be made because of the static nature of AspectJ. Secondly, the code must be recompiled each time one wants to apply the pattern to a new class, remove the pattern from some class or change the implementation of the pattern. Only programmers well versed in AOP concepts can make changes in the subject – observer relationships or change their roles. The users/non-programmers cannot make any changes in the pattern to change the system behavior.

Borella [6] provides a new solution to the observer pattern using AOP approaches. He has introduced an Agent aspect in between the subject and observers whose role is to observe the subject and execute some action on behalf of the observer each time the subject state changes. It tends to solve the first problem by enabling runtime changes in the dependencies between the observer and subject. However, the other problems persist which not only reduce the flexibility of the solution but also make the solution dependent on AOP approach and languages.

In order to employ the benefits of AOP and also to keep the solution independent of it, Jicheng *et al* [7] provide a novel implementation of observer pattern by aspect based on java annotations. They use java annotations instead of aspect language to define the pattern concerns. These annotated classes are then analyzed by a program to generate the aspects. This method provides a roundabout way of achieving separation of pattern concerns into aspects and hence increases the effort of the developer.

Our solution provides separation of pattern logic/concern from the main application logic and that too without depending on AOP concepts or languages. It enables runtime changes to be made in the pattern participants as well as their interdependencies. The code need not be recompiled to apply the pattern to a new class, remove the pattern from some class or change the implementation of

the pattern. Even the users/non-programmers can make changes in the pattern to change the system behavior.

3 Problems with the Observer Pattern and Pattern Usage in General

3.1 Problems with the Observer Pattern

Several drawbacks have been identified in observer pattern (adapted from [1] and [8]):

- **Implicit coupling between the observer and the subject:** The subject has all the observing objects registered with it, which need to be notified when a change occurs in the subject. Thus the subject has to maintain information regarding its observing objects.

- **Unexpected updates:** Since the observer dependency criteria aren't well defined or maintained, it may lead to spurious updates which may be hard to track down.

- **The push and pull model of updates:** The push model increases coupling because the subject must have some information about the observers to send detailed information about the change. The pull model may be inefficient because the observers must ascertain what has changed without the help of the subject.

- **Dangling references:** All the registered observers should remove themselves from the subject before getting destroyed otherwise they give rise to zombie or dangling references and a memory leak.

- **Implicit event invocation:** When there is chain of event invocations, it becomes hard to follow and debug by looking at the code.

- **Only programmers can make changes:** Consider the case of multiple observers observing multiple different subjects. Their dependency choices are encoded in their implementation code. Thus only the programmer can change the dependency criteria and that too in a non-runtime situation. After making the desired changes the classes have to be recompiled. It is not possible to make runtime changes in the dependency criteria by users/non-programmers. This can be costly and cumbersome when such changes are to be done frequently.

- **Partial observation:** It is difficult to define partial observation rules. Suppose an observer O is observing only some properties of a subject. A change occurs in the subject but the properties which are observed by O remain unchanged. Then too, O will be notified of change and will be updated. This happens because O registers itself with the subject not with its properties. This gives rise to un-necessary dependencies and hence un-necessary updates. Separate and explicit definition of partial observation rules can mitigate this problem as described in the general demographics survey information system example in the introduction section.

3.2 Problems with the Pattern Usage in General

The same classes in an application are burdened with the responsibility of implementing both the pattern logic (which is application independent) as well as the application logic (which is application specific). This reduces the reusability of the pattern at the implementation level. The code implementing the pattern logic is so much intertwined with the code implementing the application logic that it becomes almost impossible to identify the existence of a pattern in a code. The pattern eventually gets lost in the code [9] and it becomes very difficult to identify, add/remove and maintain patterns at the implementation level [10].

If multiple patterns are present in an application then it is very difficult to identify the specific instances of a pattern [9]. That is to say that, if same classes are acting as participants in multiple patterns then it becomes very difficult to identify the patterns of which they form participants. This may also give rise to cycles of mutually dependent classes which are difficult to maintain and reuse [9].

4 Ontology - Driven Observer Pattern - The Proposed Solution

This section describes the generic observer ontology and the static, dynamic and implementation aspects of the proposed solution along with its consequences.

4.1 The Generic Observer Ontology

We have built a generic ontology which we call observer ontology. The observer ontology encodes the design pattern logic of the observer pattern. Figure 1 shows the observer ontology concepts along with object properties connecting those concepts. The classes playing different roles in the pattern are made the instances of the corresponding ontology concepts. In the proposed solution, when a property of the observable is changed, all the observers depending upon that property along with their update methods, are retrieved from the ontology. These update methods are then invoked. If the different observers have different update priorities, then their update priorities are also extracted from the ontology and their update methods are invoked in order of their update priorities.

The same class can play the role of both the observer as well as the observable. That is, a class *B* which depends on some class *A* may have other classes *C* and *D* depending upon it. This implies that all *B*, *C* and *D* classes depend upon *A* and must be notified when *A* changes. Since the 'isObservedBy' object property is transitive by nature so the ontology reasoner can automatically deduce all the classes in the chain which depend upon a particular observable class. Thus the ontology object properties along with their characteristics (like the transitive nature of 'isObservedBy') and domain - range restrictions formally define the observer pattern domain.

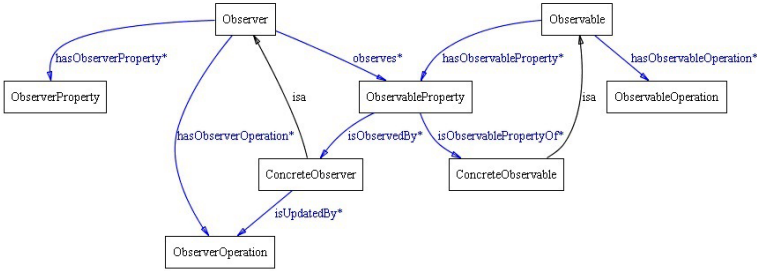


Fig. 1. Partial View of Observer Ontology Generated by the Ontoviz Plugin of Protege3.4

4.2 Structure

Figure 2 shows the structure of the ontology driven observer pattern. No dependency exists between the Observer and Observable class hierarchies. The OntologyManager class contains and exposes the notifyObservers() method to the observable classes which takes the changed property and its new value as the arguments and invokes the update() methods of the observers which depends on that changed property. The grey box denotes the element added later on. When a new observer element is added, its details are updated in the ontology and no code requires to be changed.

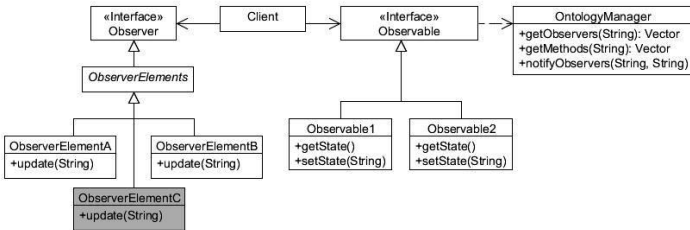


Fig. 2. Structure of Ontology Driven Observer Pattern

4.3 Collaboration

Figure 3 shows the collaboration between the classes in the ontology driven observer pattern. When a user action or any other event changes a property of an observable object, it calls the notifyObservers() method of the OntologyManager class thereby passing the property that has changed along with the new value of the property. The OntologyManager class calls its getObservers() method which queries the ontology and retrieves the observers which depend upon that particular observable property that has changed. For each observer, the update method is retrieved from the ontology by the OntologyManager and is invoked.

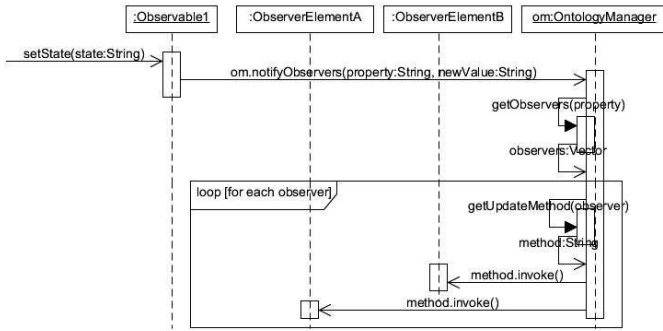


Fig. 3. Sequence Diagram of Ontology Driven Observer Pattern

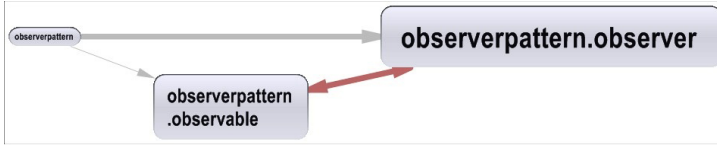
4.4 Setting Up the Ontology

Since the pattern logic is separated from the code and encapsulated in ontology, it can be changed (even at runtime) easily without modifying the code. The main idea is to enable the users (non-programmers) to directly make changes in the ontology so as to change the behavior of the application. Thus the proposed ontology driven observer pattern enables the application to be modified/extended even by non-programmers by just configuring the ontology. To edit the ontology metadata and define new observation rules, a user interface can be provided.

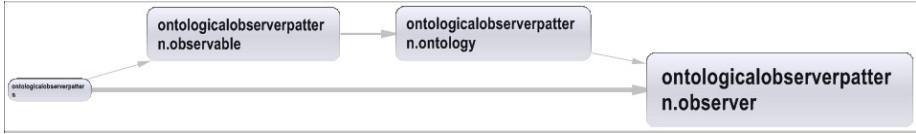
The ontology can be implemented using a database, XML or RDFS models. A specialized user interface can be created to define the ontology rules and create instances. If the ontology is implemented in RDFS and the user is familiar with some RDFS ontology editor (like Protégé) then he can directly use the editor to modify the ontology.

4.5 Analysis

We developed an example application in java based on the traditional observer pattern. We then, implemented the same application based on the ontology driven observer pattern to analyze and compare both of them. We analyzed both the applications using the JArchitect tool [11]. In figure 4(a) and 4(b) single arrow edge from A to B indicates that A is using B and double arrow edge indicates that A and B are mutually dependent. The thicknesses of edges are proportional to the strength of coupling in terms of the number of members involved. Figure 4(b) clearly shows the decoupling obtained in case of ontology driven observer pattern. In figure 5(a) and 5(b) a blue matrix cell denotes that the package in column is using that in row, a green matrix cell denotes that the package in column is used by that in row and a black matrix cell denotes mutual dependency. There are 2 black matrix cells in Figure 5(a) while no black matrix cell in Figure 5(b). This denotes the absence of cyclic dependency in ontology driven observer pattern.

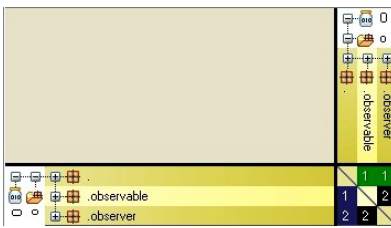


(a)

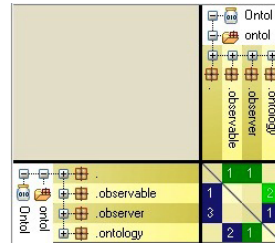


(b)

Fig. 4. Dependency Cycle Graph between the Components of (a) The Traditional Observer Pattern (b) The Ontology Driven Observer Pattern



(a)



(b)

Fig. 5. Dependency Matrix between the Components of (a) The Traditional Observer Pattern (b) The Ontology Driven Observer Pattern

4.6 Consequences

The ontology driven observer pattern has following benefits:

1. No coupling exists between the observer and the subject. The observers don't have to register themselves with the subject and hence the subject is totally unaware of its observers.
2. Since the dependency relationships between the observer and the subject are maintained explicitly so there are no spurious updates of the observers.
3. Since the notification mechanism is handled by a separate component (OntologyManager) so neither push nor pull models of updates are employed between the observer and the subject.
4. No registration and de-registration of observers is done with the subject. So no problem of dangling references can arise.

5. There is no chain of implicit event invocations so the code is clean and easy to understand.
6. The observer and observable class dependencies are maintained explicitly and declaratively in a separate ontology component. This ontology component can be provided with a GUI by means of which even the users/non-programmers can make changes in the dependency relationships even during the runtime. The participant classes of the pattern can also be changed thus resulting in the change of behavior of the entire application.
7. If some observers are observing only a few properties of some subject then this information is maintained easily and explicitly. So the proposed solution is particularly useful where multiple observers are partially observing multiple subjects.
8. The proposed solution makes it easy to identify the pattern when it exists in a code (because of the ontology component). Since the pattern logic is separated from the application logic, the pattern essentially becomes a plug and play component. The ontology component comprising of the ontology and the ontology manager is totally generic and is reusable even at the implementation level. The proposed solution makes it easy to add and remove the pattern to and from the application code.
9. Even when multiple patterns are present, the classes which form the participants of the observer pattern can be easily identified (as they are the instances of the observer pattern domain concepts). Thus it is easy to maintain and reuse the classes.

The proposed solution also has some liabilities and limitations:

1. The programming language used to implement ontological observer pattern should support reflection. Such languages include java, C#, Objective-C, PHP, Python and Ruby.
2. The use of reflection can impose a performance penalty. Therefore, worst case analysis of the software system with the pattern in place needs to be done when performance is an issue or when dealing with real-time systems.
3. Debugging can become more difficult primarily because the control flow is interrupted by the ontology component.
4. The pattern can be used by developers who are well acquainted with ontology as a structural framework for organizing knowledge.
5. The design pattern logic is totally separated from the application logic and is encoded in the ontology component. This raises an overhead of an additional component i.e. the ontology. Whenever the generic ontology is used in an application, it needs to be instantiated at least once.

5 Conclusion

The proposed ontology driven observer pattern mitigates all the identified drawbacks of the GoF observer pattern. It decouples the participant classes of the observer pattern and makes the code easy to understand. The proposed solution

enables even the users/non-programmers to make changes in the dependency relationships between the observers and the subjects even during the runtime. The participant classes of the pattern can also be changed during runtime thus resulting in the change of behavior of the entire application. The proposed solution is particularly useful where multiple observers are partially observing multiple subjects. The proposed solution makes it easy to identify the pattern when it exists in a code (because of the ontology component). Since the pattern logic is separated from the application logic, the pattern essentially becomes a plug and play component. The ontology component comprising of the ontology and the ontology manager is totally generic and is reusable even at the implementation level. The proposed solution makes it easy to add and remove the pattern to and from the application code. Even when multiple patterns are present, the classes which form the participants of the observer pattern can be easily identified.

References

1. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley (1995)
2. Gruber, T.: A translation approach to portable ontology specification. *Knowledge Acquisition* 5(2), 199–220 (1993)
3. Hachani, O., Bardou, D.: Using aspect-oriented programming for design patterns implementation. In: *Proceedings of Workshop Reuse in Object-Oriented Information Systems Design* (2002)
4. Hannemann, J., Kiczales, G.: Design pattern implementation in java and aspectj. In: *Proceedings of the 17th ACM SIGPLAN conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2002)*, vol. 37(11), pp. 161–173 (2002)
5. Garcia, A., Sant'Anna, C., Figueiredo, E., Kulesza, U., Lucena, C.J.P., von Staa, A.: Modularizing design patterns with aspects: A quantitative study. In: Rashid, A., Akşit, M. (eds.) *Transactions on Aspect-Oriented Software Development I*. LNCS, vol. 3880, pp. 36–74. Springer, Heidelberg (2006)
6. Borella, J.: The observer pattern using aspect oriented programming. In: *Proceedings of the Viking Pattern Languages of Programs, Viking PLOP* (2003)
7. Jicheng, L., Hui, Y., Yabo, W.: A novel implementation of observer pattern by aspect based on java annotation. In: *Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010)*, vol. 1, pp. 284–288 (2010)
8. Fowler, M.: *Organizing presentation logic: observer gotchas* (2006), <http://martinfowler.com/eaDev/OrganizingPresentations.html#observer-gotchas> (accessed on March 20, 2012)
9. Soukup, J.: Implementing patterns. In: Coplien, J.O., Schmidt, D.C. (eds.) *Pattern Languages of Program Design*, pp. 395–412 (1995)
10. Budinsky, F., Finnie, M., Yu, P., Vlissides, J.: Automatic code generation from design patterns. *IBM Systems Journal* 35(2), 151–171 (1996)
11. JArchitect: A tool to evaluate java code base (v3.0 2012), <http://www.javadepend.com/> (accessed on April 10, 2012)

Part V

The First International Workshop on Social Business Intelligence: Integrating Social Content in Decision Making

Towards a Semantic Data Infrastructure for Social Business Intelligence

Rafael Berlanga¹, María José Aramburu², Dolores M. Llidó¹,
and Lisette García-Moya¹

¹ Departamento de Lenguajes y Sistemas Informáticos,

² Departamento de Ciencia e Ingeniería de los Computadores,
Universitat Jaume I, Avda. Vicent Sos Baynat, S/N. Castellón de la Plana
{berlanga,aramburu,dllido,lisette.garcia}@uji.es

Abstract. The tremendous popularity of web-based social media is attracting the attention of the industry to take profit from the massive availability of sentiment data, which is considered of high value for Business Intelligence (BI). So far, BI has been mainly concerned with corporate data with little or null attention with the external world. However, for BI analysts, taking into account the Voice of the Customer (VoC) and the Voice of the Market (VoM) is crucial for putting in context the results of their analyses. Recent advances in Opinion Mining and Sentiment Analysis have made possible to effectively extract and summarize sentiment data from these massive social media. As a consequence, VoC and VoM can be now listened from web-based social media (e.g., blogs, reviews forums, social networks, and so on). However, new challenges arise when attempting to integrate traditional corporate data and external sentiment data. This paper aims to introduce these issues and to devise potential solutions for the near future. More specifically, the paper will focus on the proposal of a semantic data infrastructure for BI aimed at providing new opportunities for integrating traditional and social BI.

Keywords: Social Business Intelligence, Linked Data, Sentiment Analysis.

1 Introduction

The massive adoption of web-based social media for the daily activity of e-commerce actors, from customers to marketing departments, is attracting more and more the attention of Business Intelligence (BI) companies. So far, BI has been confined to corporate information systems, with little attention to external data. Capturing external data for contextualizing data analysis operations is a time-consuming and complex task that, however, would bring large benefits to current BI environments [10]. The main external contexts for e-commerce applications are the Voice of the Customers (VoC) and the Voice of the Market (VoM) forums. The former regards the customer opinions about the products and services offered by a company, and the latter comprises all the information related to the target market that can affect the company business. Listening to the VoM allows setting the strategic direction of a business based on in depth

customer insights, whereas listening to the VoC helps to identify better ways of targeting

and retaining customers. As pointed by [11], both perspectives are important to build long-term competitive advantage.

The traditional scenario for performing BI tasks has dramatically changed with the irruption of the Web 2.0, and the proliferation of opinion feeds, blogs, and social networks. Nowadays, we are able to listen to the VoM and VoC directly from these new social spaces to extract relevant information from their posts. This has been possible thanks to the burst of automatic methods for performing sentiment analysis over these new information sources [8]. These methods directly deal with the posted texts to identify global assessments (i.e. reputation) over target items, to detect the subject of the opinion (i.e., aspects) and its orientation (i.e., polarity). From now on, we will refer to all the data elements extracted from the opinion posts by means of sentiment analysis tools as *sentiment data*.

A good number of commercial tools have recently appeared in the market, for listening and analyzing social spaces and reviews forums, see for example Radian6 Insight, Media Miser, Scout Labs, Wise Window and Sinthesio, to mention just a few. Unfortunately, these commercial tools aim to provide customized reports for end-users, and sentiment data on which these reports rely are not publicly available (indeed this is the key of their business). Consequently, critical aspects such as the quality and reliability of the delivered data cannot be contrasted nor validated by the analysts. This fact contrasts to the high quality that BI requires for corporate data in order to make reliable decisions.

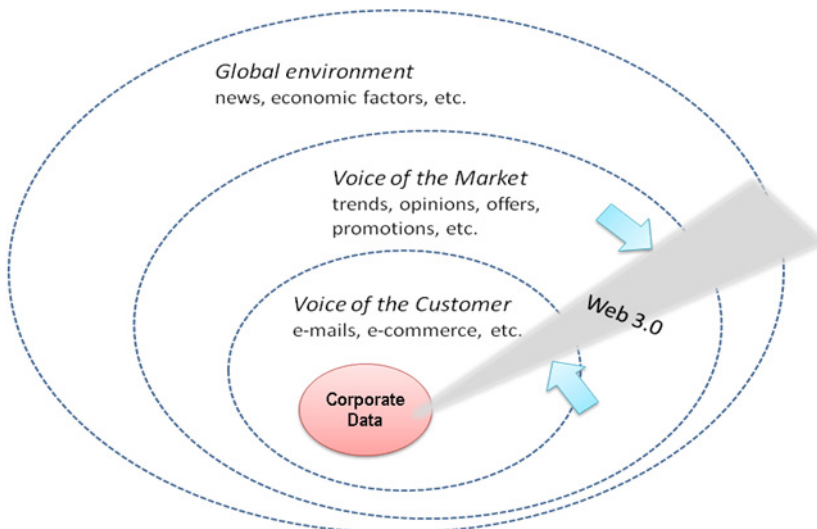


Fig. 1. Business Intelligence contexts and their relation to the Web 3.0 data infrastructure

In this paper we discuss the opportunity and advantages of defining a new data infrastructure for performing social BI. As Figure 1 shows, VoC and VoM sentiment

data must be integrated in this infrastructure together with all the external factors that may potentially affect a company business (e.g., new legislations, financial news, etc.). We claim that such a data infrastructure must follow the principles of the Linked Data initiative [6], where some preliminary proposals for BI already exist (e.g., *Schema.org* and *GoodRelations*). If existing web-based social data is migrated to the Web 3.0 as linked data in order to be shared, validated and eventually integrated with corporate data, a new global BI scenario for e-commerce applications can be achieved. Furthermore, our hypothesis is that most data and vocabularies used by researchers and companies for performing sentiment analysis could be better exploited if they were shared, contrasted and validated by the community.

2 Analytical Patterns for Social BI

Traditional BI assumes the existence of a controlled set of data sources, from which summarized data is obtained for decision making. BI architectures usually rely on a data warehouse architecture defined under a multidimensional model (i.e., just consisting of measures and dimensions) [7]. The data warehouse is then periodically fed with data extracted from the identified data sources by executing the Extraction, Transform and Load (ETL) processes. Finally, data are summarized with efficient BI tools such as Online Analytical Processes (OLAP) [2].

From a BI point of view, sentiment data extracted from social media can be also regarded as a multidimensional model. For example, the reputation of a product, the most outstanding features of some brand, or the opined aspects of an item can be represented as multidimensional data, and efficiently computed through OLAP [4].

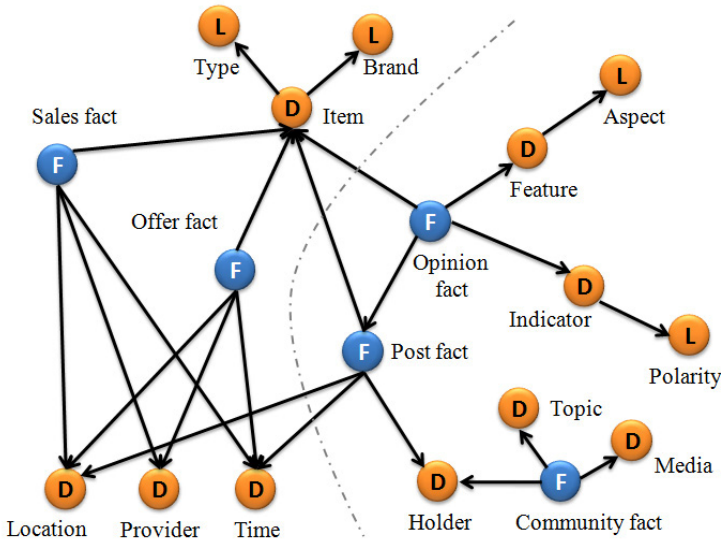


Fig. 2. Main BI Patterns in a Social analysis context scenario

The main BI e-commerce patterns we consider in this paper are summarized in Figure 2. Facts (labeled with ‘F’) represent spatio-temporal observations of some measure (e.g., units sold, units offered, number of positive reviews, and so on), whereas dimensions (labeled with ‘D’) represent the contexts of such observations. In some cases, facts can have a dual nature, behaving as either facts or dimensions according to the analyses at hand. For example, in Figure 2, a post can be either a fact or a dimension of an opinion fact. Dimensions can further provide different detail levels (labeled with ‘L’). For example, the dimension Item can be provided with the levels “Type” and “Brand”. On the other hand, facts must be associated to one or more numerical measures. For example, the sales facts are associated to units sold, promotion facts are associated to offered units, and so on.

In Figure 2, the traditional and social BI patterns are separated with a dotted line. However, the intended data infrastructure is aimed at seamlessly integrate all BI data in order to perform any kind of analysis that could require to combine corporate with external social information.

The main facts concerning social BI are *opinion facts*, *post facts*, and *community facts*. *Opinion facts* are observations about sentiments expressed by opinion holders concerning concrete aspects or features about an item, along with their sentiment indicators. For example, the sentence “I don’t like the camera zoom” express an opinion fact where the feature is “zoom”, and the sentiment indicator is “don’t like” (negative polarity). Post facts are observations of published information about some target item, which can include a series of opinion facts. Examples of post facts can be reviews, tweets, and comments published in a social network. Notice that opinion facts are usually expressed as free text in the posts, and therefore it is necessary to process these texts to extract the facts [8]. Finally, *community facts* are observations about the opinion holders that interchange sentiments about some topic. These facts are usually extracted from social networks by analyzing the structure emerged when the opinion holders discuss about some topic [12]. Notice that topic-based communities can be very dynamic as they rise and fall according to time-dependant topics (e.g., news, events, and so on).

As for the measures associated to these social facts, Table 1 shows some examples of typical measures used in the literature for sentiment and social analysis.

Table 1. Examples of measures for social BI facts

Measure	Example values	Fact type
Polarity	(-1,0,+1)	Opinion
Rating	(★,★★,★★★,...)	Post
Like	(👍,👎)	Post
Popularity	(-10,...,+10)	Community
Credibility	(0,...,10)	Community

Some examples of interesting sentiment and social analysis operations that can be done with the previous BI patterns are the following ones:

- To identify troublesome parts of some product or service.
- To measure the popularity of product promotions from rival companies.
- To find the best points in social networks to advertise a product.
- To predict the popularity of a topic in the different communities.
- To analyze the evolution of an item sentiment within a topic-based community.

3 Semantic Data Infrastructures

The Linked Data (LD) initiative aims at creating a global web-scale infrastructure for data [6]. Relying on the existing WWW protocols, this initiative proposes to publish data under the same principles that web documents, that is, they must be identified through a Unique Resource Identifier (URI), with which any user or machine can access to their contents. Similarly to web documents, these data can be also linked to each other through their URIs. In order to manage the resulting data network, data must be provided with well-defined semantics to allow users and machines to rightly interpret them. For this purpose, the W3C consortium has proposed several standards to publish and semantically describe data, mainly the Resource Description Framework (RDF) and the Ontology Web Language (OWL). In this paper we refer to *semantic data infrastructures* to the data networks resulted from publishing and linking data with the standard formats RDF and OWL.

Semantic data infrastructures provide a series of standards and tools for editing, publishing and querying their data [6]. As a result, the Linked Open Data (LOD) infrastructure serves as a main reference for semantic web researchers and developers. The basic component of this infrastructure is the *dataset*, which consists of a set of RDF triples that can be linked to other LOD datasets. These datasets usually provides a SPARQL endpoint, with which data can be accessed via declarative queries. Additionally, SPARQL also enables distributed queries over linked datasets.

This kind of global data infrastructures are opening new opportunities to both data providers and consumers to develop new applications, which goes beyond the corporate boundaries (also called data islands). More specifically, LD has opened new ways to perform e-commerce activities such as retailing, promotion, and so on. Projects like *Schema.org* and *GoodRelations* are allowing the massive publication of product offers as micro-data, as well as specific vocabularies for e-commerce applications. Additionally, commercial search engines like Google and Yandex are adopting these formats to improve the search of these data. Concerning the publication of sentiment data, recent projects like MARL [14] attempt to provide schemas for publishing user opinions as linked data.

All these projects are focused on making available product offers and reviews to end-users as well as third party applications (e.g., mobile apps). However, they are not appropriate for performing large-scale BI analyses. In the next section we identify the main requirements for a data infrastructure aimed at social BI.

4 Requirements for a Social BI Data Infrastructure

Regarding the nature of the data to be published in the data infrastructure, we have identified a set of global requirements that are not covered yet by current proposals, namely:

1. The infrastructure must give support for massive generation of opinion data from posts (e.g., reviews, tweets, etc.) so that high volumes of crawled data can be quickly processed and expressed as linked data. As in data warehouses, a series of ETL processes are needed to periodically feed the data infrastructure. These ETL processes are quite unconventional, as they deal with semi-structured web data, perform some kind of sentiment analysis, and output RDF triples.
2. Sentiment data published in the infrastructure must be semantically represented under well-controlled vocabularies and useful taxonomical relationships. Currently, sentiment data is automatically extracted from texts with either statistical [5] or Natural Language Processing (NLP) methods [8], but they do not bring well-defined semantics for enabling BI analyses. For example, most automatic methods capture “features” and “opinion indicators” from text reviews but these data is not organized into semantics groups (e.g., optics, storage and image quality for cameras) to properly calculate the partial scores for each semantic group. In this context, we have to say that the success of traditional BI partially stems from the capacity of OLAP tools for exploring data through hierarchical dimensions. Another relevant aspect to take into account is the context-dependent nature of these data [9], which can also require inference capabilities.
3. The infrastructure must support high distribution of data, providing optimal partitions w.r.t. to data usage. BI analyses are subject-oriented and consequently are focused on a given topic. Therefore, data must be distributed according to these topics. For example, opinion facts should be organized into item families (e.g., electronic products, tourist services, etc.) and allocated into different datasets. The massive data distribution also alleviates the storage requirements of these huge volumes of data.
4. The infrastructure must provide fresh data by migrating as quickly as possible published posts. Apart from the considerations of the first point, the infrastructure must adopt as much as possible the existing vocabularies in e-commerce in order to facilitate the load of data from the different sources (e.g., micro-data).
5. The infrastructure must ensure the quality and homogeneity of the datasets, dealing with the potential multi-lingual issues of a BI scenario. As e-commerce acts in a global market, sentiment data extracted from different countries will be expressed in different languages. Datasets must support multi-lingual expressions as well as organize them around semantic concepts (see second point). Additionally, links between datasets of the intended infrastructure must be as much coherent as possible, using the appropriated classes and data types offered by it. Some current approaches like MARL allow users to express opinion facts with any kind of resource (e.g., a string, a URI to an external entity, etc.) [14]. Although this makes the schema much more flexible to accommodate any opinion fact, it makes

unfeasible to perform a BI analysis over these data.

- The provided data sets are intended to be exploited by companies in order to perform analysis operations that contextualize their internal corporate data with external sentiment data. By this reason, data provided by the infrastructure must be published in a format that can be easily unloaded and integrated with corporate data in the way required by each kind of BI application.

5 A Possible Architecture

Regarding the previous requirements, Figure 3 proposes a possible architecture for the intended social BI data infrastructure. First, we divide the involved datasets into two layers. Thus, the inner ring of Figure 3 regards the main components of the proposed infrastructure, whereas the outer ring comprises the external datasets and vocabularies (dotted boxes) that are related to the infrastructure. The infrastructure components are organized according to the BI patterns shown in Section 2, and the requirements described before. Thus, each component in Figure 3 regards a different perspective of a BI analysis, and it consists of a series of topic-oriented datasets that share a common ontology.

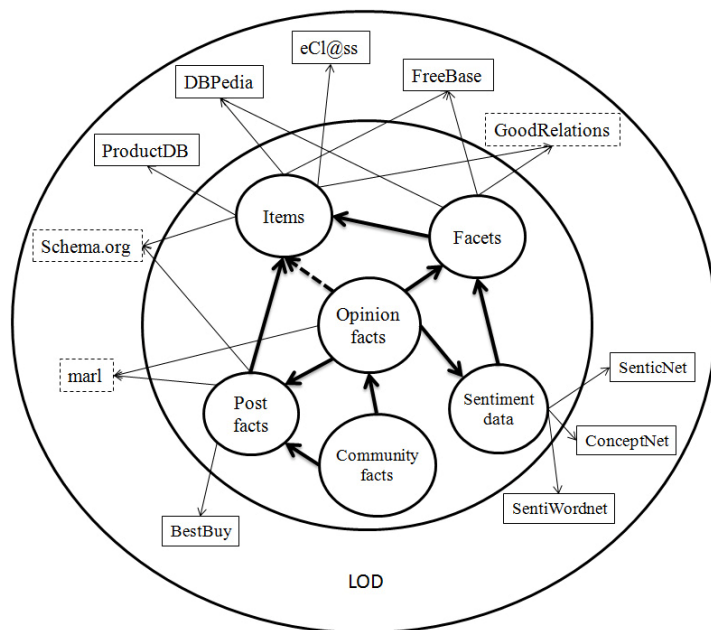


Fig. 3. Proposed architecture for the social BI data infrastructure

Links between components are considered hard links, in the sense that they must be semantically coherent, and they are frequently used when performing analysis

tasks. For example, the infrastructure should facilitate the join operations between triples of these datasets. On the other hand, links between infrastructure components and external datasets are considered soft links, as they just establish possible connections between entities of the infrastructure and external datasets (e.g., DBpedia). These external datasets are useful when performing *exploratory analyses*, that is, when new dimensions of analysis could be identified in external datasets.

A first implementation of this architecture can be found in [4] where a corporate data warehouse is enriched with sentiment data from opinion posts. In this preliminary work, linked data was not yet available. Instead, the original opinion texts were used to extract sentiment data that was stored into the corporate data warehouse. The multidimensional data model of this data warehouse included the same elements as the inner ring in Figure 3 plus some elements of the outer ring to represent hierarchies extracted from Wikipedia. With the resulting BI system it was possible to analyze sentiment and corporate data in a combined way.

6 Conclusions

We have presented a proposal for a semantic data infrastructure for sentiment data extracted from web-based social media. Its purpose is to facilitate the massive analysis of sentiment data by exploiting the ever-increasing amount of publicly available open linked data. The infrastructure components are designed to describe all necessary information for opinion analysis: products/services, features/aspects, and opinion indicators, reviews and facts and so on. The infrastructure also incorporates the functionality required to perform massive opinion analysis: the extraction of opinion facts from text reviews, and the linkage of opinion data to other datasets. This will allow the exploitation of opinion-related dimensions of analysis that are out of reach for traditional BI applications.

It is very important to highlight that one of the main advantages of using linked data is that it is semantically enriched and related. This fact ensures the applicability of data in open scenarios and facilitates its processing for many different kinds of BI applications. As for future work, our purpose is to apply semantic annotation as a key enabling technology for fast migration of opinion posts to the proposed architecture for the infrastructure. The generation of the different datasets will also rely on our previous work about unsupervised statistical sentiment analysis [4, 5].

Acknowledgements. This work has been partially funded by the “Ministerio de Economía y Competitividad” with contract number TIN2011-24147.

References

1. Cambria, E., Song, Y., Wang, H., Howard, N.: Semantic Multi-Dimensional Scaling for Open-Domain Sentiment Analysis. *IEEE Intelligent Systems* (2013), doi:10.1109/MIS.2012.118

2. Codd, E.F.: Providing OLAP to user-analysts: An IT mandate (1993)
3. Esuli, A., Sebastiani, F.: SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In: Proc. LREC 2006, pp. 417–422 (2006)
4. García-Moya, L., Kudama, S., Aramburu, M.J., Berlanga, R.: Storing and analysing voice of the market data in the corporate data warehouse. *Information Systems Frontiers*, 1–19 (2013), doi:10.1007/s10796-012-9400-y
5. García-Moya, L., Anaya-Sánchez, H., Berlanga, R.: A Language Model Approach for Retrieving Product Features and Opinions from Customer Reviews. *IEEE Intelligent Systems* (2013), doi:10.1109/MIS.2013.37
6. Heath, T., Bizer, C.: *Linked Data: Evolving the Web into a Global Data Space*, 1st edn. Morgan & Claypool, San Rafael (2011)
7. Inmon, W.H.: *Building the Data Warehouse*. John Wiley & Sons (2005)
8. Liu, B.: *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers (2012)
9. Lu, Y., Castellanos, M., Dayal, U., Zhai, C.X.: Automatic construction of a context-aware sentiment lexicon: an optimization approach. In: WWW 2011, pp. 347–356 (2011)
10. Pérez, J.M., Berlanga, R., Aramburu, M.J., Pedersen, T.B.: Contextualizing data warehouses with documents. *Decision Support Systems* 45(1), 77–94 (2008)
11. Reidenbach, R.E.: *Listening to the Voice of the Market: How to Increase Market Share and Satisfy Current Customers*. CRC Press (2009)
12. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proc. of LREC, vol. 2010 (2010)
13. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. *JASIST* 61(12), 2544–2558 (2010)
14. Westerski, A., Iglesias, C.A.: Exploiting Structured Linked Data in Enterprise Knowledge Management Systems. An Idea Management Case Study. In: Proc. EDOCW, pp. 395–403 (2011)

Subjective Business Polarization: Sentiment Analysis Meets Predictive Modeling

Caterina Liberati and Furio Camillo

¹ DEMS, Università di Milano-Bicocca, piazza Ateneo nuovo 1, 20126 Milan, Italy

² Department of Statistics, Università di Bologna, via Belle Arti 41,
40122 Bologna, Italy

Abstract. The growth of Internet and the information technology has generated big changes in subjects communication, that, nowadays, occurs through social media or via thematic forums. This produced a surge of information that is freely available: it offers the possibility to companies to evaluate their credibility and to monitor the "mood" of their markets. The application of Sentiment Analysis (SA) has been proposed in order to extract, via objective rules, positive or negative opinions from (unstructured) texts. Communication literature, instead, highlights how such polarization derives from a subjective evaluations of the texts by the receivers. In business applications the receiver (i.e. marketing manager) is led by the values and the mission of the company. In our paper we propose a strategy to fit brand image and company values with a subjective SA, a probabilistic Kernel classifier has been employed to get discrimination rule and to rank classification results.

Keywords: Text Polarization, Semantic Technology, Kernel Discriminant, Business Solution.

1 Introduction

The importance of word-of-mouth has been widely stated in a large part of the marketing literature [2],[13]. The concept is always been related to the perceptions of a product/service. Anderson (1998) stated that unsatisfied customers do engage in greater word-of-mouth than satisfied ones. Therefore, it cannot be ignored when a business wants to check its reputation in the market or wants to test the customer satisfaction of its products. Similarly, Goldenberg et al. (2001) claimed that consumers decision making process is strongly influenced by word-of-mouth. As such, word-of-mouth communications at the micro-level can influence macro-level phenomena [24]. Recently, the unprecedented growth of Internet and the information technology has generated big changes in subjects communication that, more and more frequently, occurs through social media or via thematic forums. This has resulted in a surge of information that is freely available online in a text format. For example, many online forums and review sites exist for people to post their opinions about a product [9]. While such consumer-generated content offers possibility to businesses to evaluate their

credibility and to monitor the "mood" of their markets, some concerns arise, due to complaints or smear campaigns. In such context, in fact, prompt and accurate understanding of sentiments expressed within the online text could lead to effective marketing strategies and service recovery and could help the management to identify key drivers for improved customer relationships. A big problem, indeed, is represented by the mounting frustration to discern, among thousands of pieces of data, which are the most relevant. The process of accessing all these raw data, heterogeneous both for source, type, protocol and language used, transforming them into information, is therefore inextricably linked to the concepts of automatic textual analysis and synthesis, hinging greatly on the ability to master the problems of semantic interpretation. Therefore, researchers and marketing people have been paying much attention to Sentiment classification and analysis [21],[25]. Sentiment Analysis (SA), also known as Opinion Mining, is the extraction of positive or negative opinions from (unstructured) text [18]. It analyses texts, divided into sentences, to detect their polarity and to aid the simultaneous assessment of positive and negative strength of the opinions collected. Several studies have shown a positive impact in efficient marketing response in a wide range of applications as product comparisons, opinion summarizing, and reason mining [16], [22], [20]. In order to provide fast and efficient decision processes several Machine Learning (ML) algorithms have been combined with the Sentiment. For example, Abbasi et al. (2008) proposed the Support Vector Regression Correlation Ensemble (SVRCE) approach to analyze emotional states. Pang et al. (2002) investigated several supervised ML methods to semantically classify movie reviews, Camillo et al. (2006) have shown a Kernel Discriminant classifier for predicting customers purchase propensity of a web fashion portal, Chen et al. (2011) presented a neural network based approach to distinguish comments from blogs, just to cite a few. Most of the times, in literature review, we spotted 2 issues never addressed: 1) The application of text Sentiment has been pushed, in data mining works, in order to provide objective rules for extracting people's opinions. On the contrary, Communication literature highlighted that text polarization derives from a subjective evaluations by the reader. The texts are recognized and interpreted in the light of personal values and of the environment in which the reader lives. Although SA has been introduced for supplying such polarization, it is not able to address such choice by itself, in fact, it needs of the user intervention to adjust results. Such picture is really common in business context: texts are not negative or positive regardless, their Sentiment have to be computed according to the brand values and to the market environment decided by the business management. This leads companies to define subjectively what include in a positive communication. Such communication has to be rebuilt via Sentiment Analysis. 2) Classification of the SA results could be realized via a discriminant model which assigns every instance to a group of text polarization (positive, negative, neutral). In literature, most of the Machine Learning algorithms employed, are hard classifiers. A better solution would be the application of a probabilistic discriminant that is able to detect not only the texts Sentiment, but also to assign a likelihood values to each classification.

Such an approach is more robust and reliable and provide easy-to-rank results. In our paper we would like to take into account those two aspects. We present a strategy to rebuilt via an intelligent Semantic Lemmatization the information collected in the texts coming from different web sources: forum, blogs and social networks. The Sentiment is not computed via an algorithmic rule but it has been provided by means of a subjective evaluation of the web posts. Then, a probabilistic Kernel classifier has been employed to get the discrimination. Such an approach first, better responds to the receiver's demands in terms of coherence, second it improves the classification respect to other parametric models as Gaussian Linear Discriminant Analysis. We tested our strategy on a real case of study: we analyzed data referred to an international "beauty" group present all over the world. The rest of this paper is organized as follows: section 2 provides an outline of the modeling process, recalling properties and main features of the tools employed. In Section 3, we present the data, the objective of the case of study and the model assessment. Section 4 illustrates results of our approach, finally, conclusions and future considerations about the further developments in Semantic classification are discussed in Section 5.

2 Methodological Framework

This section describes the methodology used through-out this study. Section 2.1 illustrates how to extract knowledge from unstructured texts. Semantic Analysis applied to the corpus collected, provides an intelligent way to rebuild the informative landscape present in the data. Section 2.2 gives an overview of the Kernel Discriminant Analysis, with a particular focus to the model under the multivariate Gaussian distributional assumption. This allows a probabilistic derivation of a non linear classification: predictions are reliable if they exceed a threshold of minimum probability.

2.1 Semantic Approach to a Corpus

Textual data analysis is an explorative technique which extracts information from texts. Getting knowledge is not an automatic process: first a parsing has to be performed that allows to individuate tokens (or words) of the corpus then the conversion of words in part of speech (based on their syntactic category) leads to highlight more relevant texts. Finally in order to reduce the corpus dictionary, all word variations are merged into a single representative form (lemmatization). The result is a high-dimensional document-by-term matrix where each cell in the matrix represents the raw frequency of appearance of a term in a document. Such matrix, usually, is very sparse i.e. it contains a lot of zeros since not all documents contain all corpus terms. In order to reduce the dimensionality of the feature space two ways are feasible: 1) computing a Single Vector Decomposition (SVD) to the matrix considering it as a numeric matrix. 2) computing factorization employing Correspondence Analysis (CA). We choose the second way because CA [4], [11], [15] is a factorial technique that displays categorical

variables in a property space and maps their associations in two or more axes. The fundamental motivation lies in the distance metric choice which can not be linear in this context. In Correspondence Analysis a general row i of the matrix T^1 , of order $I \times J$, is considered as a point in R^J with coordinates $f_{ij} = f_i$ ($j = 1, \dots, J$) and weight f_i ($i = 1, \dots, I$); the centroid of the rows set is the point $f_{.j}$ ($j = 1, \dots, J$). The proximities between rows are measured using the χ^2 distance. So, the square distance between rows i and i' is defined as follows:

$$d^2(i, i') = \sum_{j=1}^J \frac{1}{f_{.j}} \left(\frac{f_{ij}}{f_i} - \frac{f_{i'j}}{f_{i'}} \right) \quad (1)$$

Consistently with the geometric approach, the dispersion of the set of rows (and symmetrically, of the set of columns) around its centroid is measured through the inertia:

$$\phi^2 = \sum_{i=1}^I f_i d^2(i, \text{centroid}) = \frac{\chi^2}{n} \quad (2)$$

being n the total counted units. CA inspects the distances between row profiles as a whole or equivalently, the distances between each profile and the mean profile and describes the discrepancy from the independence model by displaying approximations between rows on the axes of maximum dispersion (principal axes). The principal axes search can be obtained by performing a Principal Component Analysis (PCA) on the table \tilde{T} whose general term is shown below:

$$\tilde{t}_{ij} = \frac{f_{ij} - f_i \cdot f_{.j}}{f_i \cdot f_{.j}} \quad (3)$$

Therefore, the vectors of the row scores are

$$\mathbf{x}_s = \tilde{T} D_J u_s = \sqrt{\lambda_s} v_s \quad (4)$$

where λ_s and v_s are eigenvalues and eigenvectors of the matrix $\tilde{T}' D_I \tilde{T} D_J$, D_I is the diagonal matrix with general term f_i , D_J is the diagonal matrix with general term $f_{.j}$.

2.2 A Probabilistic Kernel Discriminant

Kernel-based methods such as Support Vector Machines (SVMs) have been successfully used for solving various classification and pattern recognition problems for supervised learning in Machine Learning [10] [5]. A classifier in the Feature Space \mathcal{F} can be obtained via the estimation of the class conditional

¹ The two-way contingency table T ($I \times J$) is a relative frequency matrix whose general term is obtained just divided each cell values for the total sample units $f_{ij} = \frac{n_{ij}}{n}$. We denote the i -th row margin term as $f_i = \sum_{j=1}^J f_{ij}$ and the column margin is $f_{.j} = \sum_{i=1}^I f_{ij}$.

density functions and the use of Bayes’s rule. We consider the input data set $\mathcal{I}_{XY} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ of training vectors $\mathbf{x}_i \in \mathcal{R}^p$ and the corresponding values of $y_i \in \{1, 2\}$ be sets of indices of training vectors belonging to the first $y = \textit{“negative”}$ and the second $y = \textit{“positive”}$ Sentiment class, respectively. We use the nonlinear mapping

$$\Phi : \mathcal{R}^p \rightarrow \mathcal{F} \tag{5}$$

of the data from the Input Space to a higher dimensional Feature Space defined by a *kernel function* $k(x, y) = \phi(x)\phi(y)$ [17]. Let $f_g(x)$ denote the class-conditional density of the kernelized data in a generic class and the correspondent prior probability π_g , such that $\sum_{g=1}^2 \pi_g = 1$.

We classify an observation to a class for which the posterior probability of group membership is the greatest. This is achieved by utilizing the Bayes’s rule or theorem [3]. We assumed that the class conditional density is a multivariate Gaussian model in \mathcal{F} given by:

$$f_g(x) = \frac{1}{(2\pi)^{n/2} |\Sigma^\Phi|^{1/2}} \exp \left\{ -\frac{1}{2} (\Phi(x) - \mu_g^\Phi)' (\Sigma^\Phi)^{-1} (\Phi(x) - \mu_g^\Phi) \right\}. \tag{6}$$

Thus, we obtain the log posterior probability of group membership just applying the Bayes’s rule:

$$\log P(Y = g \mid X = x) \simeq \log f_g(x) + \log(\pi_g) \tag{7}$$

$$= -\frac{1}{2} (\Phi(x) - \mu_g^\Phi)' (\Sigma^\Phi)^{-1} (\Phi(x) - \mu_g^\Phi) + \log(\pi_g) \tag{8}$$

So, comparing the two classes (1 and 2), we assign the observation vector \mathbf{x}_i to class 2 if

$$D_{g=1}(\mathbf{x}_i) > D_{g=2}(\mathbf{x}_i) \tag{9}$$

where the generic kernel discriminant function is

$$D_g(x) = \log(\pi_g) - \frac{1}{2} (\Phi(x) - \mu_g^\Phi)' (\Sigma^\Phi)^{-1} (\Phi(x) - \mu_g^\Phi) \tag{10}$$

If the maximum in 10 does not uniquely define a class assignment for a given observation vector \mathbf{x}_i , we can then use random assignment to break the tie between the appropriate classes, or we can decide to assign only those observations with a posterior probability above certain probability threshold. This algorithm can be computed by means of any matrices calculus packages, although the computation complexity is $O(n^3)$, the advantage in terms of good classification rewards the machine stress.

3 The Case and the Model Assessment

The case study analyzed in our work, is a real business case. The data refers to a international enterprise of beauty products, present all over the world. The aim of the study is related to the web reputation of the company: the estimate and the assessment of an intelligent text classifier which is able to discriminate between positive and negative web posts. Since the company is a well known brand, it does not surprise the big amount of texts coming from blogs or thematic forums we collected (50000 posts). The monitoring period of the sources of information has been about 1 month: spontaneous and solicited discussions about the usage of cosmetic products, or the issues related to products for the care and beauty of the body, have been downloaded and analyzed. Obviously, we did not model all the web posts, but we selected, randomly, a training sample of 9000 documents. Such documents have been read and classified by the company, according to its communication assets and brand mission. Then, SyN Semantic Center, a complex system of text intelligence analytical-linguistic, produced by Synthema, (an Italian company of Human Language Technology), has been performed. SyN Semantic Center is an advanced technology platform that allows to run linguistic and semantic analysis of any piece of information, such as documents, web pages, discussion groups, forums, chats, e-mails, databases, scientific and technical publications. By means of sophisticated linguistic analysis functions, this platform can detect the key elements of any text, according to different criteria: morphology, syntax, logics, and semantics. In our case, first it classified each word from a grammatical point of view (morpho-syntactic analysis), in order to reduce the number of concepts described. Secondly, through a logic-functional analysis it identified who is doing what, how, when and where. Finally, via a semantic analysis the platform interpreted the underlying meaning of each single word.

Therefore, an accurate semantic lemmatization, containing a reduced number of stems and concepts, is obtained and the frequency matrix $\text{texts} \times \text{conceptual-lemmas}$ (T), finally, is produced. Correspondence Analysis computed on the T allows to get the Principal Component axes (40 PCs) which composed the X matrix of our discriminant model. As we highlighted in Subsection 2.2, the kernel trick replaces the dot products in \mathcal{F} with a kernel function in the Input Space, so that the nonlinear mapping is performed implicitly in the new Space [23]. Optimal generalization of kernel-based method, still depends on the selection of a suitable kernel function and the values of regularization and kernel parameters [7]. In literature, many kernel functions are present we can choose from. The most common are shown in the Table 1. Except for the Polynomial which has as unknown parameter the degree of the transformation (here set to 2), for the other maps an estimation process has been performed, for selecting suited c values, according to the data. A grid search algorithm, for minimizing misclassification error rates and providing the c parameters with the correspondent

Table 1. Kernel Functions

Kernel Mapping	$k(\mathbf{x}, \mathbf{z})$
Cauchy (CAU)	$\frac{1}{1 + \ \mathbf{x} - \mathbf{z}\ ^2}$
Gaussian (RBF)	$\exp\left(-\frac{\ \mathbf{x} - \mathbf{z}\ ^2}{2c^2}\right)$
Laplace (LAP)	$\exp\left(-\sqrt{\frac{\ \mathbf{x} - \mathbf{z}\ ^2}{c^2}}\right)$
Multi-Quadric (MULTIQ)	$\sqrt{\ \mathbf{x} - \mathbf{z}\ ^2 + c^2}$
Polynomial (POLY)	$(\mathbf{x} \cdot \mathbf{z})^2$

errors, has been employed. Then, the best kernel discriminant “fed” a Nearest Neighbors (NN) with a window width $\delta = 9$.

4 Results

In this section we illustrate our results on the training sample, using the model strategy that can be outlined in the following steps:

1. Getting the target variable y via a subjective Sentiment provided by the company
2. Getting the matrix X performing CA on the matrix T document \times lemmas
3. Performing Probabilistic Kernel Discriminant (with different kernel map)
4. Obtaining discriminant function for each Kernel map
5. Choosing the best map via ROC curve comparison
6. Hybridizing Kernel function with k-nearest neighbor adjustment

Hybridization consists in applying a Nearest-Neighbor to the kernel discriminant got at 5). Such method is used to generate a nonparametric density estimate in each group and to produce a classification criterion²[14].

The strategy has been repeated for all the 5 Kernel maps shown in the Table 1, but it can be easily adjust to any other function. Before illustrate the results, we have to point out that the sample selected presents a severe overbalance in the 2 Sentiment classes: the negative comments ($y=1$) weigh for 24% of the overall dataset, therefore, the positives ($y=2$) weigh for the leftover 76%. We expect that such class distribution imbalance will affect the misclassification error rates especially that one of the Linear Discriminant. Visual inspection of the Figure 1 reveals how much any kernel map improves the classification.

² In our hybridization a NN with uniform weights and $\delta = 9$ has been employed. This generated a posterior probability of groups membership for each document.

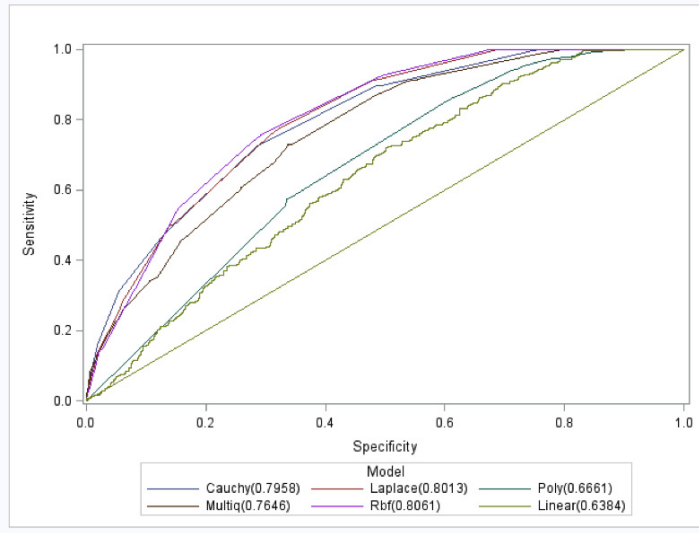


Fig. 1. ROC curves comparison

In particular, if we use the RBF we obtain the greatest area under its ROC curve (0.8061) among the five choices. It allows to gain over 26% in prediction respect to the use of the standard Linear Discriminant. Finally, the hybridized solution obtained using the RBF discriminant as input of a NN provides the best calibrated score: its misclassification error rate is the minimum, but it shows, also, a good prediction in both classes (Table 2).

Table 2. Confusion matrix with the Hybridized RBF Discriminant

		into y	
		1	2
from y	1	75.84	24.16
	2	29.20	70.08
priors		0.50	0.50

5 Conclusions

Sentiment Analysis has been massive applied in the recent years. This is due to the exponential growth of the information freely available, as blogs and web forums where people use express their opinions. Such picture makes the analysis of the unstructured texts particularly interesting also for the businesses. Monitoring the customers' "mood" and the brand reputation become two important assets to adjust marketing communication. This explains the importance

of providing a robust classifier which is able to discriminate the Sentiment of the subjects posts. In this paper we presented a strategy for documents classification with respect to a subjective polarization of Sentiment. The innovative aspect of this strategy consists in the application of Kernel Discriminant Analysis to the Principal Components of a Lexical Correspondence Analysis performed on a matrix texts-forms, stemming from a sophisticated lexical pre-processing of a large set of web posts. The fundamental idea underlying our approach lies in the fact that text Sentiment is related to a subjective interpretation of the reader. This interpretation is strictly connected to cultural issues and mind modeling. In order to manage such complex sentiment polarization, our proposal is to perform predictive models based on sophisticated lexical and statistical approaches.

A valuable extensions of this work would be to compare our solution with other classification models, and testing the results via cross-validation, to get confidence intervals of the error rates.

References

1. Abbasi, A., Chen, H., Thoms, S., Fu, T.: Affect Analysis of Web Forums and Blogs using Correlation Ensembles. *IEEE Transactions on Knowledge and Data Engineering* 20(9), 1168–1180 (2008)
2. Anderson, E.W.: Customer Satisfaction and Word of Mouth. *Journal of Service Research* 1, 5–17 (1988)
3. Bayes, T.: Studies in the History of Probability and Statistics: IX. Thomas Bayes' Essay Towards Solving a Problem in the Doctrine of Chances. *Biometrika* 45, 296–315 (1763/1958)
4. Benzècri, J.: Analyse des Données, Dunod, Paris. Analyse des correspondances, vol. 1, (Data analysis, vol. 1:Correspondence analysis) (1973)
5. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, New York (2006)
6. Camillo, F., Liberati, C., Neri, F.: e-CRM, web semantic propensity models and micro-data mining an application of Kernel Discriminant Analysis to the Glam on web case. In: *Proceeding of 8th International Conference on Textual data Statistical Analysis, JADT 2006*, pp. 235–243. Presses Universitaires de Franche-Comt (2006)
7. Cawley, G., Talbot, N.L.C.: Efficient approximate leave-one-out cross-validation for kernel logistic regression. *Machine Learning* 71(2-3), 243–264 (2008)
8. Chen, L.S., Liu, C.H., Chiu, H.J.: A neural network based approach for sentiment classification in the blogosphere. *Journal of Informetrics* 5, 313–322 (2011)
9. Coussement, K., Van den Poel, D.: Improving customer complaint management by automatic email classification using linguistic style features as predictors. *Decision Support Systems* 44, 870–882 (2008)
10. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press (2004)
11. Escofier, B., Pagès, J.: *Analyses factorielles simples et multiples; objectifs, méthodes et interprétation*, Dunod, Paris (1988)
12. Greenacre, M.: *Theory and Applications of Correspondence Analysis*. Academic Press, New York (1984)
13. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters* 12(3), 211–223 (2001)

14. Hand, D.J.: Kernel Discriminant Analysis. Research Studies Press, New York (1982)
15. Lebart, L., Morineau, A., Piron, M.: Statistique exploratoire multidimensionnelle. Dunod, Paris (1998)
16. Lee, C.M., Narayanan, S.S., Pieraccini, R.: Classifying emotions in human-machine spoken dialogs. In: Proceedings of ICME, Lausanne, Switzerland, pp. 737–740 (2002)
17. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions Royal Society A 209, 415–446 (1909)
18. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundation and Trends in Information Retrieval 2, 1–135 (2008)
19. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86 (2002)
20. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. Journal of Informetrics 3(1), 143–157 (2009)
21. Subasic, P., Huettnner, A.: Affect Analysis of Text Using Fuzzy Semantic Typing. IEEE Transaction on Fuzzy Systems 9(4), 483–496 (2001)
22. Tang, H., Tan, S., Cheng, X.: A survey on sentiment detection of reviews. Expert Systems with Applications 36, 10760–10773 (2009)
23. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
24. Ye, Q., Zhang, Z., Law, R.: Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. Expert Systems with Applications 36, 6527–6535 (2009)
25. Wu, C.H., Chuang, Z.J., Lin, Y.C.: Emotion recognition from text using semantic labels and separable mixture models. ACM Transactions on Asian Language Information Processing 5(2), 165–183 (2006)

Sentiment Analysis and City Branding

Roberto Grandi¹ and Federico Neri²

¹ Alma Mater – Università di Bologna, Bologna, Italy
roberto.grandi@unibo.it

² Synthema, Semantic Intelligence, Pisa, Italy
federico.neri@synthema.it

Abstract. The Web is a huge virtual space where to express and share individual opinions, influencing any aspect of life, with implications for marketing and communication alike. Social Media are already an important marketing arena.

This paper describes, on one hand, the characteristics of Sentiment Analysis and, on the other hand, the results of its application to an empirical research on the city of Bologna and on its brand perception on the Web.

In the international scenario a growing number of cities compete with each other in order to attract: investors and foreign companies; different types of tourists, and new residents.

City branding can be considered the starting point for developing effective policy of city marketing. The Bologna City Branding Project aims at increasing the effectiveness of territorial marketing policies carried out by the municipality of Bologna.

This study partially confirms and partially rejects what many sectors of the city would have expected from the perception of Bologna on the Web. From the point of view of academic research, it has shown the potential of Sentiment Analysis in the study of perception of the city brand. Further investigations should be made to integrate this approach with the more qualitative and quantitative techniques. From the point of view of the place marketing of cities, the results of this research have shown that place marketing is a complex activity and that, in order to be more effective, an integrated plurality of approaches have to be promoted and used.

Keywords: city branding, sentiment analysis, text mining, lexical analysis, semantic analysis, opinion mining, unsupervised clustering, semantic role labeling.

1 Introduction

The Web is a huge virtual space where to express and share individual opinions, influencing any aspect of life, with implications for marketing and communication alike. Reviews and ratings on the Internet are increasing their importance in the evaluation of products and services by potential customers. In certain sectors, it is even becoming a fundamental variable in the “purchase” decision. Internet users often evaluate products or services online. Consumers tend to trust the opinion of other consumers, especially those with prior experience of a product or service, rather than

company marketing. Social Media are influencing consumers' preferences by shaping their attitudes and behaviors. The influence of the Internet, especially via social networking, on people's purchasing behavior has grown over the years. Monitoring the Social Media activities is a good way to measure customers' behavior, keeping track of their sentiment towards brands or products, of the impact of campaigns and the success of marketing messages, identifying and engaging the top influencers who are most relevant to the brand, product or campaign. Social media are already an important marketing arena.

These factors have led to a burgeoning industry with a plethora of companies offering Sentiment Analysis services in Social Media. Sentiment Analysis and Opinion Mining are established, although nascent, fields of research, development and innovation. The goal is always broadly the same; to know "who" is speaking about "what", "when" and in "what sense".

Opinion Mining and Sentiment Analysis are important for determining opinions on brands and services, or understanding consumers' attitude. Given the relentless cascade of information on the Internet, in the last decade the field of automatically extracting opinions has emerged, being not possible to keep up with the flow of new information by manual methods [1]. There is a large body of work on Opinion Mining for English, not for Italian, by automatic means [2][3]. Globally, two techniques are used: Supervised Machine-Learning [3][4] and Unsupervised methods, that use a lexicon with words scored for polarity values such as neutral, positive or negative [5]. Supervised methods require a training set of texts with manually assigned polarity values and, from these examples, they learn the features (e.g. words) that correlate with the value. Chaovalit and Zhou [6] evaluated common implementations for both techniques on movie reviews and concluded that Supervised techniques perform with about 85% accuracy, whereas Unsupervised methods perform about 77%.

Besides the computational technique that is used for Opinion Mining, there is a whole gamut of issues that play a role in the quality and usability of the opinion extraction. First of all, opinion mining can be applied to different levels of text: words, phrases, sentences, paragraphs or documents. Words, as the smallest units, can have different polarities in different meanings and or in different domains. This requires word sense disambiguation of words in context and domain, or topic detection as prior processing [7]. Furthermore, polarity expressed by a word may be reversed within a phrase through negation. Also, parts of a document may express different polarities.

This paper describes, on one hand, the characteristics of Sentiment Analysis and, on the other hand, the results of its application to an empirical research about the brand perception of the city of Bologna on the Web. This research has been carried out as part of the Bologna Branding Project, aiming at measuring the presence of the city of Bologna on Social Media between the end of 2012 and March 2013.

2 City Branding

In the international scenario a growing number of cities compete with each other in order to attract: investors and foreign companies; different types of tourists and new residents, such as students, skilled workforce or talents [8] [9] [10] [11][12][13][14].

Today, in order to compete in the place marketing, it is necessary to develop a policy of city branding.

From the communication point of view, the brand can be considered as a means which “builds meaning”. Companies need to design communication strategies able to increase the value of brands, starting from their identity [15] [16][17]. As part of a process of corporate brand Zenker and Braun [18] defines place brand as “... *a network of associations in the consumers’ mind based on the visual, verbal, and behavioral expression of a place, which is embodied through the aims, ..., and the general culture of the place’s stakeholders and the overall place design*”. In this theoretical framework, the city branding satisfies rational, functional, symbolic and emotional needs [12].

For these reasons city branding can be considered the starting point for developing policy of city marketing, which can be implemented in the frame of the city’s unique proposition[19] which characterizes its positioning. From this point of view “*city branding is understood as the means both for achieving competitive advantage in order to increase inward investment and tourism, and also for achieving community development, reinforcing local identity and identification of the citizens with their city and activating all social forces to avoid social exclusion and unrest*” [20].

In recent years, the city branding has become an important research domain that “*has been the subject of constant debate between several contrasting academic disciplines which have studied the phenomena of city branding with different methods*” [21].

In our opinion the first stage of a process of city branding is to analyze the perception of the city brand by different audiences, both internal and external. The perceptions of the audiences “*are a stronger determinant of positive or negative outcomes, and so measuring these perceptions in place of ‘real’ characteristics seems to be more valuable and meaningful-even though place identity is unquestionably one key driver of place perception*” [22].

3 Bologna City Branding Project

3.1 Bologna City Branding Project

Bologna is an Italian city with 375,000 inhabitants and home for the oldest university in the world. Bologna is worldwide renowned for its well preserved historic center, for the high level of cultural consumption, for the quality of its food. Bologna is not characterized for a single tangible excellence, even though it possesses numerous excellences. It is not considered until today an important tourist destination. The Bologna City Branding Project – coordinated by the Bologna Urban Center - aims at increasing the effectiveness of territorial marketing policies carried out by the municipality of Bologna. This objective is pursued by defining the positioning that the city wants to achieve.

This project spreads through various stages. From October 2012 to March 2013 was carried out the analysis of the perception of the brand image of Bologna locally, nationally and internationally. In the second phase the positioning of the city of

Bologna will be determined, taking into account the perceived image by the different target groups and the strategic plan of the city. Later it will be carried out a public competition for the Bologna logo and payoff. The next step will be the definition of the communication strategy and the elaboration of the communication plan.

3.2 Perception of the City of Bologna Brand by Different Target Groups

In the opinion of Sebastian Zenker [23] the main ways to measure the perceptions of the brand of the city are three. The first method has the form of free brand associations of target audiences with qualitative method, the second has the form of *“attributes uncovered with quantitative methods like standardized questionnaires on different brand dimensions”*. The third method is characterized by a use of mixed methodologies: such as *“multidimensional scaling, network analyses”* and others. In this paper we will present the results for the third study, that adopted the Sentiment and Knowledge Mining approach to measure the presence of the city of Bologna on Social Media.

4 The Sentiment Mining System

The Sentiment and Knowledge Mining system [24][25] used in this study is built on the following components:

4.1 The Crawler

The crawler is a multimedia content gathering and storing system, whose main goal is managing huge collections of data coming from different and geographically distributed information sources. The crawler provides default plug-ins to extract text from most common types of documents [24].

4.2 The Semantic Engine

This component identifies the relevant knowledge from the whole raw text, by detecting semantic relations and facts in texts. Concept extraction is applied through a pipeline of linguistic and semantic processors that share a common knowledge. The shared knowledge base guarantees a uniform interpretation layer for the diverse information from different sources and languages.

4.2.1 Lexical and Semantic Analyses

The automatic linguistic analysis of the textual documents is based on Morpho-Syntactic, Semantic, Semantic Role and Statistical criteria. At the heart of the lexical system is the McCord's theory of Slot Grammar [26][27]. The system analyzes each sentence, cycling through all its possible constructions. It tries to assign the context-appropriate meaning – the sense - to each word by establishing its context. The parser

- a bottom-up chart parser - employs a parse evaluation scheme used for pruning away unlikely analyses. It builds the syntactical tree incrementally. By including the semantic information directly in the dependency grammar structures, the system relies on the lexical semantic information combined with Semantic Role relations. The detected terms are then extracted, reduced to their Part Of Speech (Noun, Verb, etc) and Semantic Role (Agent, Object, etc) tagged base form and used as documents metadata [24].

4.2.2 Sentiment Analysis

The Sentiment Analysis is based not only on the polarity of words, but also on the syntactical tree of the sentence being analyzed. The system identifies idiomatic expressions, giving interpretation to negations, modifying polarity of words basing on the related adverbs, adjectives, conjunctions or verbs, in particular taking in account specific functional-logic complements [28][29].

4.3 The Search Engine

Users can search documents by Natural Language queries, expressed using normal conversational syntax, or by conceptual keywords, or by combining concepts into SAO (Subject-Action-Object) triples [25]. Reasoning over facts and semantic structures makes it possible to handle diverse and more complex types of questions. Traditional Boolean queries in fact, while precise, require strict interpretation that can often exclude information that is relevant to user interests. By mapping a query to concepts and relations very precise matches can be generated.

4.4 The Classification Engine

The automatic classification of documents is made by Unsupervised Clustering. The application dynamically discovers the groups of documents which share some common traits.

5 Collecting the Data

Around 20,000 posts and blog contributions related to Bologna have been collected by focus crawling techniques. The system has gathered both pages coming from selected and reliable sources of information and generic textual contributions .

6 Navigating the Data

6.1 The Space of Concepts

6.1.1 The Concept Cloud

A Tag Cloud is a visual representation for text data, typically used to quickly perceive the most prominent terms in a set of documents. The Concept Cloud instead uses the

- c) The perceived image of a city is determined by both tangible and intangible characteristics. The foreigners appreciate the kindness (“*wonderful*”, “*kind*”) and the “*accent*” of local people, although - sometimes - they are “*too many*” (Fig. 3). The visitors appreciate the “*home family*” atmosphere of the city, too.
- d) The perception of the atmosphere of a city is affected by how the tourists feel when they walk into town alone. They like to “*wander*” in the “*small*”, “*winding*”, “*narrow streets*” without a clear destination (Fig. 3). This wandering is facilitated by the structure of the historic city center (“*a charming town with narrow streets*”, “*wandering the streets without the need to make real plans*”). The streets of Bologna are considered “*crowded*” and “*dirty*”, too.
- e) Foreigners associate Bologna to music and musicians. They listen to the numerous live concerts, which are held in the streets and in the city clubs (“*we wandered the streets, ... listened to music, yelled, sang, danced...*”). This association legitimizes the appointment of the city as *City of Music* by Unesco.
- f) The presence of verbs, adjectives and adverbs, such as “*find*”, “*share*”, “*love*”, “*like*”, “*different*”, “*incredible*”, “*interesting*”, “*also*”, associated to Bologna has been influenced by a generally shared image of the city, characterized by its not being considered a major tourist destination. The fact that Bologna is not perceived as a touristic city lowers expectations and increases the chances that visitors will be positively surprised. Visitors often emphasize their pleasant surprise to discover an unknown city. Bologna is perceived as several cities in the same city and it is considered “*really*” “*also*” something else. Bologna is perceived as a “*hidden jewel*” (Fig. 4) and the magic of its arcades (“*porticos*”) are a secret “*to share*” (“... *Bologna is a hidden secret and a great example of what Italians call «the good life»*”).
- g) Generally speaking Bologna is described as the capital of the so-called Italian Motor Valley. Until now, this recognition has always been considered an important feature of the perceived image of Bologna shared by all. Surprisingly, foreigners have never associated the city of Bologna to brands such as Lamborghini and Ducati on the Web.
- h) The overall evaluation of the network users on the city of Bologna is positive, with more favorable opinions by foreign visitors

The presence of different perceptions will result in defining different strategies of city branding.

8 Conclusions

In this paper we have presented the results for the presence of the city of Bologna on Social Media between the end of 2012 and March 2013, by using a Sentiment and Knowledge Mining approach.

In this study Knowledge Mining has proved to be a powerful methodology to find associations difficult to detect with other models of analysis. It has helped on identifying the most significant *tangible* feature for the city of Bologna, that is its

porticos. So the city branding of the city will privilege the horizontal dimension than the vertical one, which would have prevailed if visitors had preferred the “*Two Towers*”. From the semantic point of view, the prevalence of the vertical dimension (Towers, Historical Buildings, Skyscrapers) focuses on the sense of sight: we look at the towers and skyscrapers. We, however, cannot see all 40 km of arcades. The porticos are not observed from the outside, but they are public spaces where people wander and like to “*get lost*”. Under the arcades we use all five senses, not just the sense of sight. Choosing the horizontal dimension of the porticos has profound implications for both the place marketing and the storytelling of the city.

Additional findings relate to the *intangible* characteristics of the city: Bologna is perceived on the Web as a friendly city, and one of the city capitals of the food and music in Italy.

Compared to the prior knowledge has been detected a novelty among the foods perceived as distinctive for the city. In addition to the traditional Bolognese dishes was reported ice cream. This indication is an important confirmation of the efforts made by the city to increase the quality of its ice cream and to be associated with the culture of ice cream, thanks to the Carpigiani Gelato Museum.

A final result refers to the supposed reputation of the city of Bologna as one of the capitals of the so called Italian Motor Valley. The city of Bologna has never been associated with brands like Lamborghini and Ducati on the Web. This result will have important implications on the place marketing of the city strategies.

The results of this research within the Bologna City Branding Project frame give rise to two recommendations. From the point of view of academic research, it has shown the potential of Knowledge Mining in the study of perception of the city brand. From the point of view of the place marketing of cities, this research have shown that place marketing is a complex activity, that, in order to be effective, must promote the use of an integrated plurality of approaches.

References

1. Sentiment Analysis Symposium 2011, New York (April 12, 2011)
2. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2(1-2), 1–135 (2008)
3. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: *Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing*, vol. 10, pp. 79–86 (July 2002)
4. Socher, R., et al.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: *Proceedings of EMNLP 2011 - the Conference on Empirical Methods in Natural Language Processing*, pp. 151–161 (2011) ISBN: 978-1-937284-11-4
5. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics* 37(2), 267–307 (2011)
6. Chaovalit, P., Zhou, L.: Movie review mining: A comparison between supervised and unsupervised classification approaches. In: *Proceedings of the Hawaii International Conference on System Sciences, HICSS* (2005)

7. Esuli, A., Sebastiani, F.: SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In: Proceedings of LREC-2006, Genova, Italy (2006)
8. Anholt, S.: Branding places and nations. In: Anholt, S. (ed.) Brands and Branding, pp. 213–226. Bloomberg Press, New York (2004)
9. Florida, R.: The Rise of the Creative Class. Basic Books, New York (2004)
10. Florida, R.: Cities and the Creative Class. Routledge, New York (2005)
11. Grandi, R.: Le città creative. *Il Mulino* 6(10), 1037–1044 (2010)
12. Grandi, R.: Cultural Planning, City Marketing and Creative Cities: Bologna – from cultural city to creative city? In: Jenei, A. (ed.) Communication with the Public from the Local Government Perspective, pp. 111–132. Corvinus University of Budapest Press, Budapest (2012)
13. Hospers, G.-J.: Creative Cities in Europe: Urban Competitiveness in the Knowledge Economy. *Intereconomics*, 260–269 (September/October 2003)
14. Kavartzis, M.: From city marketing to city branding: Towards a theoretical framework for developing city brands. *Place Branding* 1(1), 58–73 (2004)
15. Zenker, S.: Who’s Your Target? The Creative Class as a Target Group for Place Branding. *Journal of Place Management and Development* 2(1), 23–32 (2009)
16. Zenker, S.: How to catch a city? The concept and measurement of place brands. *Journal of Place Management and Development* 4(1), 40–52 (2011)
17. Ferraresi, M.: La marca. Costruire un’identità rafforzare un’immagine, Roma, Carocci (2003)
18. Grandi, R., Miani, M.: L’impresa che comunica. Come creare valore in azienda con la comunicazione, Novara, Isedi – De Agostini (2006)
19. Semprini, A.: Marche e mondi possibili, Milano, Franco Angeli (1993)
20. Zenker, S., Braun, E.: Branding a city – a conceptual approach for place branding and place brand management. In: Paper Presented at the 39th European Marketing Academy Conference, Copenhagen, June 1–4, p. 3 (2010)
21. Ashworth, G.J., Voogd, H.: Selling the City: Marketing Approaches in Public Sector Urban Planning, London, Belhaven (1990)
22. Kavartzis, M.: From city marketing to city branding: Towards a theoretical framework for developing city brands. *Place Branding* 1(1), 70 (2004)
23. Lucarelli, A., Berg, P.O.: City branding: a state-of-the-art review of the research domain. *Journal of Place Management and Development* 4(1), 12 (2011)
24. Zenker, S.: How to catch a city? The concept and measurement of place brands. *Journal of Place Management and Development* 4(1), 43 (2011)
25. Zenker, S.: How to catch a city? The concept and measurement of place brands. *Journal of Place Management and Development* 4(1), 4–46 (2011)
26. Neri, F., Pettoni, M.: Stalker, A Multilanguage platform for Open Source Intelligence. In: Open Source Intelligence and Web Mining Symposium, Proceedings of 12th International Conference on Information Visualization, July 8–11, pp. 314–320. IEEE Computer Society, LSBU, London, UK (2008) ISBN:978-0-7695-3268-4
27. Neri, F., Geraci, P.: Mining Textual Data to boost Information Access in OSINT. In: Open Source Intelligence and Web Mining Symposium, Proceedings of 13th International Conference on Information Visualization, IV 2009, July 16–17, pp. 427–432. IEEE Computer Society, Barcelona (2009) ISBN: 978-0-7695-3733-7
28. McCord, M.C.: Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars. *Natural Language and Logic* 1989, 118–145 (1989)

29. McCord, M.C.: Slot Grammars. *American Journal of Computational Linguistics* 6(1), 31–43 (1980)
30. Neri, F., Geraci, P., Camillo, F.: Monitoring the Web Sentiment”, the Italian Prime Minister’s case. In: 2010 International Conference on Advances in Social Networks Analysis and Mining (2010), 978-0-7695-4138-9/10 © 2010 IEEE Computer Society Winner of Best poster Springer Award. Odense (DK), 9-11/08/2010, doi:10.1109/ASONAM.2010.26
31. Neri, F., Aliprandi, C., Camillo, F.: Mining the Web to Monitor the Political Consensus. In: Wiil, U.K. (ed.) *Counterterrorism and Open Source Intelligence. Lecture Notes in Social Networks*, vol. 2, Springer, Wien (2011), doi:0.1007/978-3-7091-0388-3-19

A Case Study for a Collaborative Business Environment in Real Estate

Nicoletta Dessì and Gianfranco Garau

Università degli Studi di Cagliari, Dipartimento di Matematica e Informatica
Via Ospedale 72, 09124 Cagliari, Italy
{dessi, garaug}@unica.it

Abstract. According to recent vision of Web 2.0, this paper explores the prospective of implementing a business environment that enables users to be more agile in capturing and evaluating information about real estate offers. A cloud infrastructure hosts the business environment and introduces commercial services in a web community made up of a set of actors (i.e. citizens, enterprises, professionals, companies etc.). Users explore, change and share both quantitative and spatial information by means of a social network, the common venue within which they interact. Being offered as a cloud service, the business environment supports efficient and scalable data management of loosely structured information that is captured from web resources. A prototype is presented that provides users with the geographic representation of real estate offers and related statistics about the price trend.

Keywords: Business environments, cloud services, spatial data management.

1 Introduction

Recent vision of Web 2.0 technology [1] is promoting the use of flexible organizational systems (i.e. social networks, blogs and wikis) that have been introduced into current work practices. Leaders in different fields, many private sector companies promote their business on line. Seeking new ways to reach consumers, companies enrich their offers with multimedia content (i.e. photos, maps, animations, etc.) and advertise users of social networks. These initiatives are a clear symptom of the fact that Web 2.0 technologies are extending their influence to business environments and make it rational to suppose the gradual appearance of entirely new categories of business environments that will draw on Web 2.0 technological trends such as new styles in user interaction, geo-referenced data, diffusion of mobile devices, etc.

As it happens for collaborative Business Intelligence solutions [1], the development of these business environments is challenging because it calls for flexible data models to describe business entities, efficient and scalable data store to realize fast response time, methodologies for the datastore population in pay-as-you-go fashion and an integrate collaborative environment to discuss results [1]. Aiming at face this challenge, we present a case study that explores the prospective of implementing an innovative

business environment that is concerned with the evaluation and/or the choice of a property in real estate market. Our goal is to empower users to be able to manage and visualize real estate offers in a computational environment characterized by web navigation, collaboration and content of many types, including maps with the option to overlay reference information about a property or the price trend in selected regions.

We choose real estate domain because the choice of a property to buy is concerned with ascribing meaning to both quantitative and spatial information such as the cost of property, proximity to commercial centers, proximity to educational services, proximity to health centers, just to cite a few.

A distinctive aspect of our proposal is the adoption of a social perspective: we consider the users as they are organized in a social structure made up of a set of actors (such as citizens, enterprises, professionals, companies, organizations etc.). We envision that these actors focus their activities inside the business environment which provides additional access to new ideas, opinions and opportunities and enables users to be more agile in capturing information about business offers.

Our approach poses important challenges concerning the design and implementation of such an environment as it aims to offer functionalities that are not longer locked to a consolidate infrastructure as it happens in data warehousing and decision support systems, but concentrates on introducing business services in a web community while reducing IT dependencies and allowing users to analyze data, share their business knowledge and strategies in a collaborative manner. To accomplish these goals, our key idea is to use a cloud infrastructure for hosting the business environment and a social network as a common venue within which users interact.

Although your environment is devoted to a specific business domain, its architecture is quite general and can be adopted for implementing similar business solutions in other domains.

The paper is organized as follows. Section 2 presents motivations and related work. Section 3 describes the business environment architecture. Section 4 presents some functionality of the prototype we implemented. We conclude our paper in section 5, with a summary of our work.

2 Motivations and Related Work

Many reasons make real estate a representative domain for implementing the new paradigm of business environment we propose.

First, during last five years, we have witnessed a gradual process of “digitalization” of real estate supply. Previously reported in newspapers and magazines, property advertises are now published in real estate specialized web sites by web enterprises that act as brokers and provide information with multimedia content such as photos, maps, animations, etc.). The aim of these websites is not selling immediately what they offer. Instead, they want to inform users about the characteristics of their commercial offers.

Second, supported by a traditional web site, the absence of user interaction comes to be inadequate when the user decision is influenced by both quantitative and

qualitative aspects that must be evaluated within a geographical context as it happens for real-estate properties whose evaluation strongly depends both on the geographical context and user needs [2]. For example, if an area is located along a highway, this geographical feature will be appreciated by an enterprise which deals his products on a daily basis while it is a negative feature for a buyer who intends to use the area for housing purposes.

Additionally, the potential buyer tries to know the selling price of similar areas in the same region, looks for real estate offers in different web sites, consults urban plans, etc. As well it would be useful, both for real estate professionals and private citizens, to know the market trends within a specific urban area or the medium price of the houses offered in this area by various competitors.

In the past two decades, the use of spatial information technologies, especially GIS, had widely assisted managers in their work. However, the sophisticated nature and the cost of GIS often exclude many potential stakeholders or professionals from getting benefit from the use of distributed spatial information [3]. Additionally, despite their huge capacities in storing and managing geographical data, GISs have some limits in solving most real-world decision problems [4].

Numerous publications have investigated issues (e.g. data exchange, software, model sharing) of implementing decisional supports on the web and concluded that web technologies will have an huge impact on future developments [4] [5].

Similar to our approach, [6] presents the advantages of a hybrid service and process repository as the foundation for a structured marketplace for arbitrary services which not only holds a flat list of services, but also exposes a generic set of uses cases. The paper stresses the difficulties in integrating several cloud services as computing resource vendors keep their own interfaces. For solving questions arising from these heterogeneous environments, the authors envision a marketplace, a set of resources or services, where consumers can select from a variety of available services to build complex applications.

With the aim to contribute a new architectural template for heterogeneous, distributed information systems, [7] proposes a flexible service oriented architecture for planning and decision support for an environmental information management. The architecture uses real time geospatial datasets and 3D presentations tools, integrated with added-value services for environmental modelling and support decision making in case of emergency. The paper presents a case study on a forest fire crisis management system.

3 The Business Environment Architecture

Our business environment is grounded on two twin cloud applications, namely NESSIE (Network-based Environment Supporting Spatial Information Exploration) and MyNESSIE and a Datastore. Accessible by a web site, NESSIE includes advanced data management and administrative functionality. Accessible by a social network [8], myNESSIE is a canvas application that includes user-oriented assets. The objective is to help users to be informed about real estate offers through the visualization of both their location and a set of socio-economical data.

Managed by IT managers, NESSIE feeds data from external and local resources, encapsulates and manages captured information as a Dataspace [9], a paradigm for data integration intended for the management of heterogeneous data coming from diverse resources regardless their schema and location. MyNESSIE provides access to data abstracting from technical concepts such as data format or schemas and facilitates the integration of existing information through the interactive use of maps.

The Datastore is an object-oriented database that lives in the cloud and easily moves data, applications, and services in and out of the cloud. Each object belongs to a single class, which categorizes the objects, and a key that uniquely identifies it within its class. A specific class of objects, namely the class “House” is devoted to detail economic (i.e. price, typology, etc.) and geographic information (i.e. address, location, geo-tags, etc.) about real estate properties.

The Datastore is automatically populated in pay-as-you-go fashion by capturing and pre-processing data from different web sources (in our prototype, from public web sites of real estate agencies) or from local resources, including municipalities, cadastral offices, chambers of commerce, etc.

To extract data, IT managers specify a filter which details the geographic zone (region, state, city, location) and the kind of real estate property (apartment, area, building, etc.) to be selected. Then, NESSIE subscribes a service, captures and parses data and automatically feeds the Datastore. The class “Url” memorizes the addresses of the web sites from which information has been captured. As data may have different shortcomings (i.e. they are often partially structured or miss important information such as the geo-tag of a property) they are initially staged in a buffering area, checked and finally stored in the Datastore. As well, NESSIE managers are allowed to manually insert into the Datastore information about a new real estate property and content from spreadsheets in CVS format.

Stored as objects belonging to the classes “Zone” and “Macro-zone”, Google maps are the geographic areas that users define and access. They make it easy to visualize geo-referenced objects and allow users to spend less time in acquiring information of interest. Special zones (for example cadastral zones or detailed user areas) can be imported from external files. In this case, zones are first stored in the buffering area, then validated by the NESSIE administrators and finally stored in the Datastore.

Conceptually, our environment can be thought as an integrated set of application services which implement mechanisms for:

- (1) abstracting the complexity of integrating data regardless of their format and location;
- (2) providing query and navigation support with high level of flexibility to the user needs;
- (3) allowing users to
 - satisfy their information needs in an intuitive manner, simply by drawing on a map the geographic area of interest;
 - discuss their choices, share their knowledge and take decision on the purchase of a property in a collaborative fashion.

Services rely on lightweight Application Program Interfaces (APIs) and mash-ups are a flexible way for their customized composition.

4 The Business Environment Deployment

This section presents the functionalities of a prototype that implements the proposed business environment for supporting information diffusion about real estate offers in San Francisco (USA). Our prototype captures data from public web sites of agencies that offer real estate deals, RSS services and some APIs to access their contents. Additionally, the prototype captures data about the trend of real estate offers in the last four years.

The user is provided with a geographic representation of real estate offers, relevant statistics about the price trends and options for searching the objects in the Datastore. Users can visualize single maps, define and store geographical zones that are highlighted with different colors. Finally, they are enabled to aggregate different zones in a single zone, namely a macro-zone.

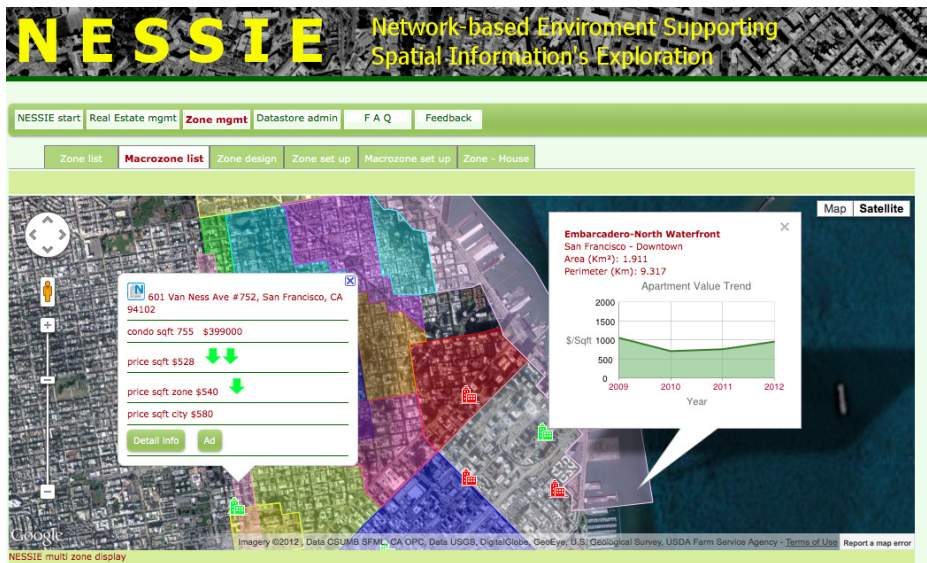


Fig. 1. NESSIE: a macro-zone in San Francisco (USA)

Fig. 1 shows how NESSIE presents information about the macro-zone “San Francisco - Downtown”. The spot on the right gives information about a specific zone (“Embarcadero-North Waterfront”) in which the user is interested. Specifically, the spot shows the area and the perimeter of the macro-zone and details the trend of the apartment value (i.e. the price per square feet) during the last four years. Real estate properties are highlighted with different colored graphic symbols. When the user clicks on a symbol, it produces the spot on the left-side of Fig. 1 which shows the address of the property, its area and cost as well statistical information about the average price of real estate properties in the zone where the property is located and in the metropolitan area of San Francisco. The down green arrows mean that the price of the property is lower than the prices in this zone and in the metropolitan area of San

Francisco. If the property does not satisfy his requirements, the user will point at a new location on the map in the same or in a different zone.

Selecting a property in a pre-fixed zone or macro-zone may eventually lead to the proposal of offers whose prices are compared to the price in the same zone. Usually, these zones are cadastral areas as defined by the urban plans. It is not unlikely that the user wants to acquire information on a new zone that crosses many cadastral zones. He can do it by defining interactively a customized zone of arbitrary form, namely a “dynamic spatial context”, and performing the following steps:

- (1) Obtain the Google map of the zone of interest;
- (2) Define interactively a polygonal bounding on the map;
- (3) Storing this map as an object belonging to the class “Zone”;
- (4) Query the Datastore for geographic and statistic information about the real estate properties the polygonal bounding geo-references;
- (5) Detail information for geo-referenced properties.

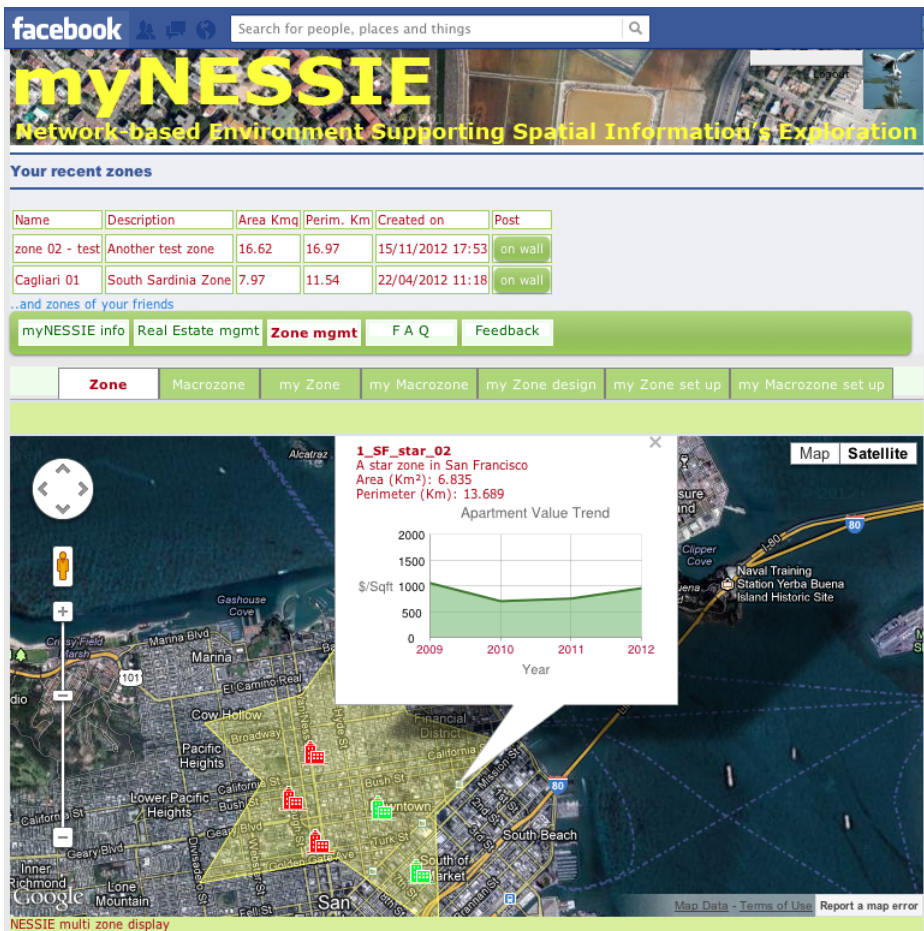


Fig. 2. MyNESSIE: example of dynamic spatial context

Even though the above steps are similar to the usual steps to obtain information on Google maps, the step (2) introduces a dynamic user interaction into the process the user is carrying on. Specifically, the user defines a polygonal bounding of arbitrary size and form (i.e. it is not necessary a regular polygonal form) simply by clicking on the map the vertices of the polygonal bounding he wants to draw.

As an example, Fig. 2 shows a dynamic spatial context where the user draws interactively a polygonal bounding that is shaped like a star. Visualized information is only about this context: the green houses represent properties whose cost is lower than the medium cost of the similar properties offered in the polygonal bounding while the red houses represent properties whose cost is higher.

Dynamic spatial contexts are innovative because there is little evidence of use of such geo-processing functions, in both cloud computing [10] and collaborative tasks [3], even though recent work [11] has shown that cloud computing can handle geo-processing tasks.

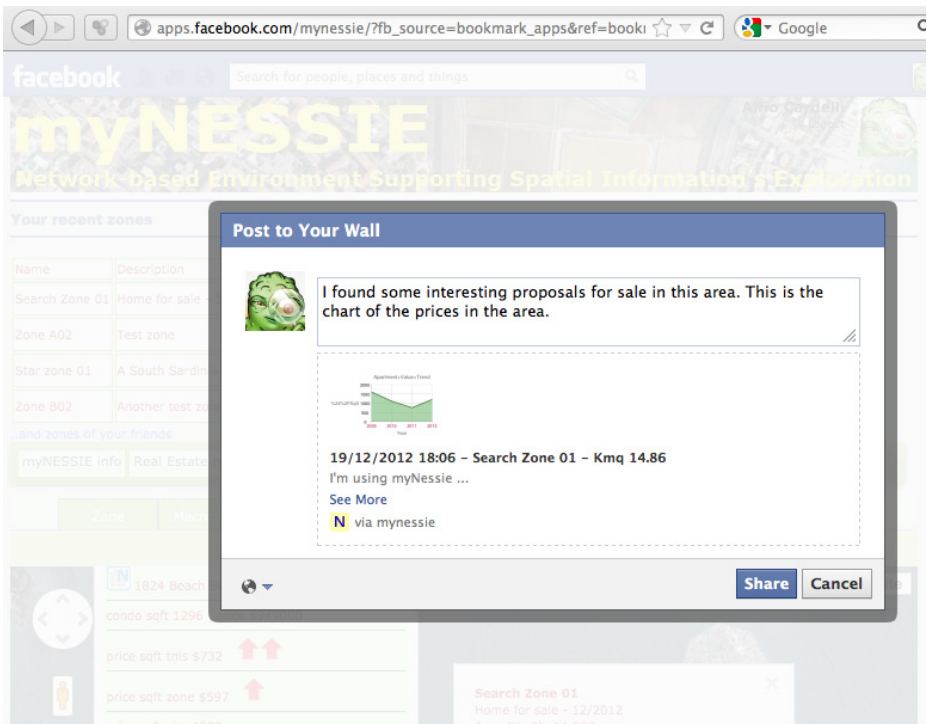


Fig. 3. MyNESSIE: user notification message

By means of Facebook, users are allowed to store permanently and share maps and spatial contexts (i.e. the maps, the polygonal bounding and the overlaid reference information) with their friends as we will explain in what follows.

Additionally, Fig. 2 lists the zones the user shares with his MyNESSIE friends. We note that these zones are not necessarily in San Francisco, but in different geographical

areas which the user and his friends are interested in. As previously mentioned, collaboration happens by means of Facebook whose users are enabled to create lists for staying in touch with other users and organize their friends as they like. To access MyNESSIE, a Facebook user is required to make a subscription that stores, into an object of the class “User”, data about the subscriber and the list of his friends. After their subscriptions, users are allowed to share maps and objects only with their friends who, in turn, subscribed MyNESSIE. As Fig. 3 shows, they can see more of them in their news feed and get notified each time new MyNESSIE objects are posted.

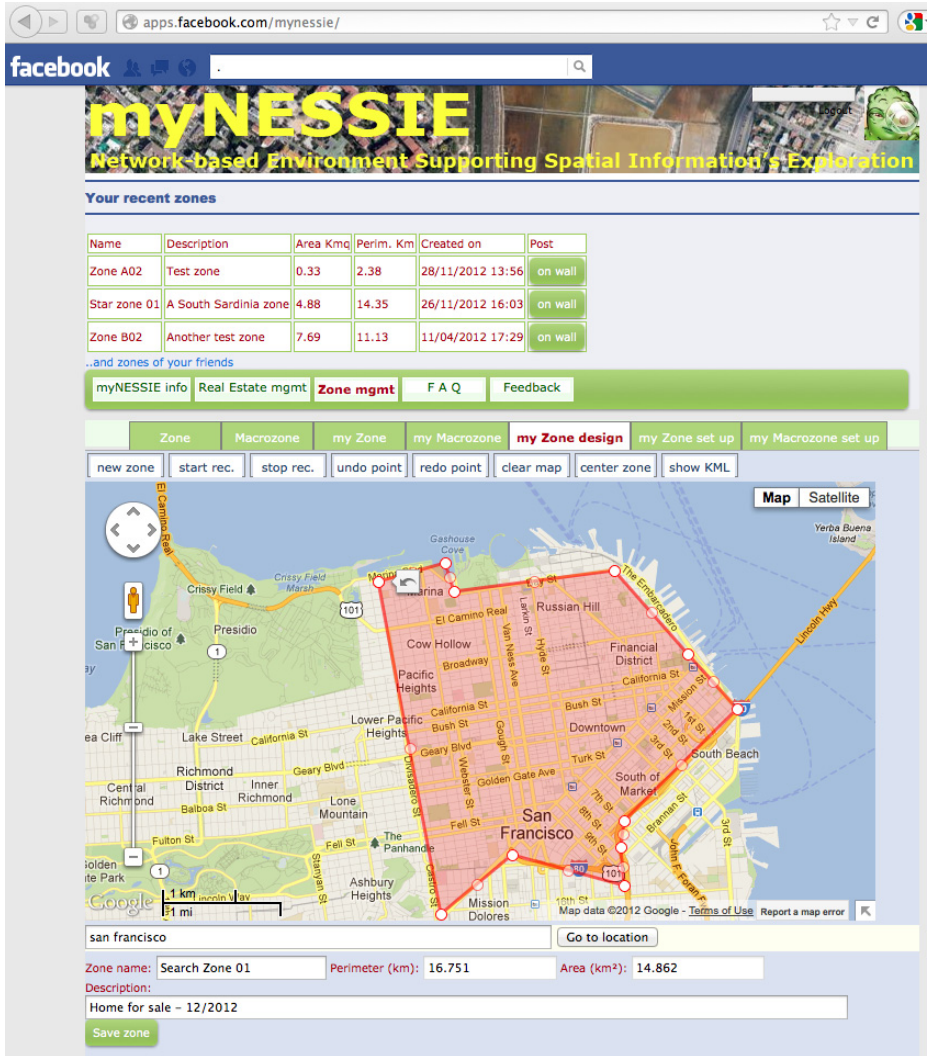


Fig. 4. MyNESSIE: additional example of user-defined zone

Different kinds of maps are available. Fig.4 shows an additional example of user-defined zone on a topological map.

Technical details. The business environment has been implemented on top of Google App Engine (GAE) [12], a cloud computing environment which provides a platform-as-a-service (PaaS) for developing and hosting web applications. It provides storage for web applications and uses a distributed architecture to automatically manage scaling to very large data sets allowing both NESSIE and MyNESSIE to maintain high performance as they receive more traffic. GAE makes available a no-relational database, namely the GAE Datastore which supports operation to access objects (i.e. create, read, update, delete) and an SQL-like language called GQL. For implementing applications, we used JavaScript/AJAX/jQuery and Django, a high-level Python web framework that runs within GAE. A limited amount of GAE resources is provided for free and this was enough for developing and running our prototype. We choose Facebook as it releases an application development platform and provides an API that allows third party applications to be integrated in it. Geo-services are supported by Google maps.

5 Conclusions

We presented a case study that explores the prospective of implementing a collaborative business environment in real estate and supports geographically referenced information. The proposed environment is flexible and easy-to-use. Users explore information about real estate properties in an interactive way, change and refine their preferences, perform personal evaluations in a real-time manner and share information by means of a social network. Devoted to a specific business domain, our case study aims to trace a road for the deployment of data management applications and collaborative business solutions.

The deployment on an emerging Cloud platform enables information services for business users and collaborative information sharing over high-volume data sources. The presented prototype exhibits the following technical requirements. First, a flexible data model describes business entities (i.e. the real estate properties) and loosely structures contextual information coming from diverse web sources. Being offered as cloud service, the Datastore supports efficient and scalable data management and is populated in pay-as-you-go fashion. Finally, the business environment allows users to easily retrieve and share both quantitative and spatial information.

Most importantly, as business environments increasingly incorporate related technologies (i.e. OLAP, Data Warehousing, Web Services, Business Intelligence, etc.) which only provide rudimentary collaboration capabilities, we tried to identify the nature of the technology we need in order to promote the development of collaborative and specialized computational solutions to support decisions in collaborative manner involving customers, professional and domain experts.

Future work will concentrate on extending the approach envisioned in this paper. Future steps include both the definition of a decisional model, the acquisition of social and economic data, the integration of additional functionality.

Acknowledgments. We are very grateful to anonymous reviewers for the useful comments and suggestions.

This research was supported by RAS, Regione Autonoma della Sardegna (Legge regionale 7 agosto 2007, n. 7), in the project “*DENIS: Dataspaces Enhancing the Next Internet in Sardinia*”.

References

1. Berthold, H., Rosch, P., Zoller, S., Wortmann, F., Carenini, A., Campbell, S., Bisson, P., Strohmaier, F.: An Architecture for ad-hoc and collaborative business intelligence. In: Proceedings of the 2010 EDBT/ICDT Workshops, pp. 1–6. ACM (2010)
2. Marchi, G., Argiolas, M.: A GIS based technology for representing and analyzing real estate values. In: Urban and Regional Data Management: UDMS 2007, pp. 345–356. CRC Press, Taylor & Francis, London (2007)
3. Rinner, C., Keßler, C., Andriulis, S.: The Use of Web 2.0 Concepts to Support Deliberation in Spatial Decision-Making. *Computers, Environment and Urban System* 32(5), 386–395 (2008)
4. Sugumaran, V., Sugumaran, R.: Web-based Spatial Decision Support Systems (WebSDSS): Evolution, Architecture, Examples and Challenges. *Communications of the Association for Information Systems* 19, Article 40 (2007)
5. Sugumaran, R., DeGroote, J.: *Spatial Decision Support Systems: Principles and Practices*. CRC Press, Taylor & Francis (2011)
6. Vigne, R., Mangler, J., Schikuta, E., Rinderle-Ma, S.: A structured marketplace for arbitrary services. *Future Generation Computer Systems* 28, 48–57 (2012)
7. Vescoukis, V., Doulami, N., Karagiorgou, S.: A service-oriented architecture for decision support systems in environmental crisis management. *Future Generation Computer Systems* 28, 593–604 (2012)
8. <http://www.facebook.com/applications/DSSs/20383310695>
9. Halevy, A.Y., Franklin, M.J., Maier, D.: Principles of dataspace systems. In: Vansummeren, S. (ed.) *PODS 2006*, pp. 1–9. ACM (2006)
10. Karimi, H.A., Roongpiboonsopit, D., Wang, H.: Exploring Real-Time Geoprocessing in Cloud Computing: Navigation Services Case Study. *Transactions in GIS* 15(5), 613–633 (2011)
11. Wang, Y., Wang, S., Zhou, D.: Retrieving and indexing spatial data in the cloud computing environment. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) *CloudCom 2009*. LNCS, vol. 5931, pp. 322–331. Springer, Heidelberg (2009)
12. <http://code.google.com/intl/it/appengine/docs/whatisgoogleappengine.html>

OLAP on Information Networks: A New Framework for Dealing with Bibliographic Data

Wararat Jakawat, Cécile Favre, and Sabine Loudcher

Université de Lyon (ERIC LYON 2), France

{wararat.jakawat,cecile.favre,sabine.loudcher}@univ-lyon2.fr

Abstract. In the context of decision making, data warehouses support OLAP technology and they have been very useful for efficient analysis onto structured data. For several years, OLAP is also used to analyze and visualize more complex data. Now, many data sets of interest can be described as a linked collection of interrelated objects. They could be represented as heterogeneous information networks, in which there are multiple object and link types. In this paper, we are focusing on bibliographic data. This type of data constitutes a rich source that is the starting point of research on bibliometrics, scientometrics domains. In this context, we discuss the interest of combining information networks, OLAP and data mining technologies. We propose a framework to materialize this combination and discuss the main challenges to build this framework. The basic idea is to be able to analyze various networks built from the bibliographic data representing different points of view (authors networks, citations networks...) and their dynamic.

Keywords: OLAP, Data Warehouse, Information Networks, Bibliographic Data, Data Mining.

1 Introduction

Communication systems, biological networks, transport systems, social and information systems on the web have become ubiquitous and their volume has increased every day. All these systems are networked systems and they usually consist of a large number of interacting and multi-typed objects [6]. Individual objects interact with a specific set of objects, forming large data sets, interconnected among them. Such interconnected, multi-typed networks or systems are called heterogeneous information networks [6,12]. They are extracted from the web, blogs and various kinds of online databases. For example, social networks are extracted from postings and blogs like Facebook; highway networks are extracted from transportation databases; publication author networks and citation networks are extracted from bibliographic databases like DBLP and MedPub etc.

Graphs have been widely used for modeling these networks and there have been numerous studies dealing with information networks, in many disciplines. The goal is to understand the structure and the behavior of information networks. Extracting knowledge inside large networks is a time-consuming and

complex task. Problems including ranking, clustering, classification, entity similarity search and relationship prediction in information networks have been studied [14]. Extracting knowledge from an information network could answer questions such as *what are the main topics of a set of publications?*, *who are the central entities in a community ?*, etc. Moreover, with such knowledge, it is possible to understand past events and to predict events in future.

In parallel, data warehouses and OLAP (Online Analytical Processing) could be very useful for dealing with heterogeneous information networks. Data warehouse systems support OLAP or multidimensional data analysis by building cubes to provide easy navigation, visualization and fast analysis for decision making within a vast amount of data. Users can view data through several dimensions or analysis axis and through different hierarchical levels for each dimension via OLAP operators.

In this paper, we outline some actual researches about OLAP on information networks and we present a new framework. In our framework, we want to build several networks of a given study, these networks representing different points of view of a same problem dealing with bibliographic data. Our goal is to model and build multiple networks and then to store them into a data warehouse. After that, we want to use OLAP for visualizing and analyzing networks. Besides we plan to combine OLAP and some data mining techniques in order to enrich the network analysis. In this paper we address the issues of such a new framework considering the case of scientific bibliographic data. We chose to deal with bibliographic data as a first application domain to test our ideas. This is a position paper to discuss the basis for future work.

The remainder of this paper is organized as follows. Section 2 deals with bibliographic data and their interest for different approaches. Section 3 introduces concepts about information networks and OLAP. Section 4 outlines general definitions of OLAP on information networks and related work. Section 5 presents our proposed framework and the related challenges. Section 6 is a conclusion.

2 Bibliographic Data

Bibliographic data analysis can be applied in many works in different areas. There are several objectives, including not only research evaluation, but also research evolution understanding, bibliometric analysis, etc. It could be useful for helping governments, managers and others to make their task easier such as deciding which projects or researchers should receive more support, who should be a reviewer, how to make evolve the topics of a conference or a journal over time.

Bibliographic data are extracted from online databases such as DBLP, ACM, PubMed, NCBI and etc. They collect large data about scientific publications in different domains, including information about authors (e.g. name and institutions) and details of publications (titles, conferences, keywords, published date and citations). It is thus possible to build networks such as co-authors network, citations network and so on.

The network can be represented as a graph containing nodes and edges. For example, co-authors network contains authors as nodes and co-author relationship as edges.

Bibliographic data have been used as a basic of many studies focusing on different challenges. Muhlenbach *et al.* proposed to discover research communities [10]. They proposed a graph-based clustering method in the case of conferences and authors. Different kinds of relationships are considered. Gupta *et al.* designed a clustering algorithm for network evolution [5]. Their node types in the network were papers, authors, conferences and terms. The algorithm can take into account the evolution both at the object level and at the clustering level. Huang *et al.* introduced the detection of the evolution of semantic communities extracted from article titles [7]. They constructed a word association network based on word relationships in titles. They used statistical distribution frequencies on edges to classify two communities. Deng *et al.* presented three models of expert-finding approaches considering the publications [4]. Their models included the statistic language model, the topic-based model and a hybrid model. Pham and Klamka provided a visualization using citation analysis [11]. Social Network Analysis (SNA) is used to determine clustering issues. The result is presented on clustering level. Several researches studied the databases of published papers in order to provide a tool or a user interface for monitoring and exploring these data [9,13].

3 Preliminaries

3.1 Information Networks

An information network is a large number of individual objects interacting with a specific group of objects [3,15]. Usually, an information network is visualized with a graph model. Each node represents an object or an entity such as actors in social networks, an edge or a link is a relationship between two entities.

Definition 1. A graph $G = (V, E)$ consists of V , a set of vertices or nodes and E , a set of edges. Each edge has two vertices associated with it.

There are two types of networks. In the first type, networks are homogeneous networks. They contain a single object type and a single link type such as friends networks, authors networks and movies networks. Links may include a label or a weight. In the other type, networks are composed of multiple object and link types and they are called heterogeneous networks. For example, a medical network can contain patients, doctors, disease entities and links can be “*is followed by*” or “*has contracted*”. The figure 1 shows two examples of bibliographic networks. In figure 1a the authors network is a homogeneous network where each node represents an author (*authorID*) and an edge represents a co-author relationship in one or several papers. For example, authors A and B have written three papers together in the same conference. A link with the weight 3 has been added between them. An example of a heterogeneous network is presented in figure 1b, it is an author-paper network. This network has two types of nodes:

authors and papers. There are three types of edges. The first link is “*written*” between authors and papers. The second represents co-author relationship and the last one relates papers written by the same author.



Fig. 1. Examples of bibliographic networks

3.2 On-Line Analytical Processing (OLAP)

In data warehouse systems, On-line Analytical Processing (OLAP) gives a multi-dimensional view of data by building data cubes [2]. The multidimensional model consists of facts representing by measures and dimensions. The data cube contains cells that include measures, which are values based on a set of dimensions. Dimensions can be seen as analysis axis and may be organized into hierarchies with several levels. Levels are structured attributes or not. For instance, in the example of the publications, the *time* dimension hierarchy may consist of four levels: *semester*, *year*, *decade*, *all*; the *venue* dimension hierarchy includes three levels: *support* (the name of the conference like *ICDM*, the name of the journal like *TKDE*, the name of the book, etc.), *research area* (like *databases*, *data mining*, *information retrieval*, etc.) and *all*. The dimensions are assumed to determine measures. Basically, measures can be numerical indicators which are calculated by aggregating the same dimensions of all facts.

An interestingly feature of the multidimensional model is the measure aggregation by using one or more dimensions, e.g., computing the total number of publications by each country over years. There are four classic OLAP operations: *roll-up* takes the current data and does a group-by on one dimension in order to aggregate or summarize facts; *drill-down* is the dual of the roll-up operator by giving more details; *slice and dice* reduce dimensions for taking a subset of data on its dimensions and *pivot* changes layouts for analyzing in different points of view.

4 OLAP on Information Networks

4.1 General Definitions

First, Chen *et al.* introduced Graph OLAP, a general framework for OLAP on information networks [3]. Graph OLAP is a collection of network snapshots

where each snapshot i has k informational attributes describing the snapshot and has a graph $G_i = (V_i, E_i)$. Such snapshots represent different sets of the same objects in real applications. For instance, with regard to the author-paper network of the figure 1b, *venue* and *time* informational attributes can mark the status of each individual snapshot e.g. *ICDM 2008* and *ASONAM 2010*; *authorID* is a node attribute defining each node, and collaboration frequency is an edge attribute reflecting the connection strength of each edge. Dimension and measure concepts, found in traditional OLAP domain, should be re-defined for Graph OLAP.

At first, there are actually two types of graph OLAP dimensions. The first one is an informational dimension, and it uses an informational attribute. These dimensions have two roles: organizing snapshots into groups based on different perspectives and granularity (each group corresponds to a cell in the OLAP cube) and controlling snapshot views but they do not touch the inside of any individual snapshot. For example, the two informational attributes *venue* and *time* with their respectively hierarchical concepts $\{semester, year, decade, all\}$ and $\{support, research\ area, all\}$ can be used as informational dimensions. We can look at the snapshot of each group e.g., $(ICDM, all\ years)$ and $(data\ mining\ area, 2010)$.

The second type of dimension is a topological dimension coming from the attributes of topological elements. Topological dimensions operate on nodes and edges within individual networks. Let us consider author network for instance, the following hierarchy $\{institute, country, continent, all\}$ associated with the node attribute *authorID* can be used for merging authors from a same institute into a generalized node. A new graph with generalized nodes is generated by summarizing the original network. In our example it shows interactions among institutions.

There are two kinds of measures in Graph OLAP. The first one is a graph. Graph is both viewed as a data source and as a special kind of measures. The second kind of measure is not a graph. It could be a node count, average degree, centrality etc. Due to different types of dimensions in graph OLAP, there are different semantics for aggregation. Let us consider an aggregated graph measure for example, aggregating data with informational dimensions groups among the snapshots such as collaborations between authors in the same conferences and during a period of time. Users can *roll-up* on the papers and grouped them by research areas. Whereas aggregating data with topological dimensions groups elements inside individual networks such as a new generalized network from author network is generated in order to have an institution network.

After this general framework proposed by Chen *et al.*, we propose a comparison between traditional OLAP and Graph OLAP (see table 1). Traditional data warehouses focus on the storage and data retrieval in contrast of data warehouses over graphs that are interested in representing information networks which are interrelated and multi-typed. Traditional data cubes take facts and generate aggregate measures. Graph cubes consider both attributes and structures for network aggregation. A given network as input is changed into a new network as

output. Two types of dimension have been presented in Graph OLAP (informational and topological dimension) whereas there is only one type in traditional OLAP. In term of measures, traditional OLAP has numeric measures and aggregation functions such as COUNT and SUM to summarize multiple records. There are two types of measures in Graph OLAP. First, the measure can take the form of a graph and the aggregation function is then specific to graph. The second type of measure is not graph but can be indicators coming from graph theory such as average degree and diameter.

In traditional OLAP there is only one semantic for operators such as roll-up. The OLAP semantics accomplished through informational dimensions and topological dimensions are different and Chen *et al.* speak about informational OLAP (abbr. I-OLAP) and topological OLAP (abbr. T-OLAP), respectively. With roll-up in informational OLAP, snapshots are just different observations of the same underlying network, and they are grouped into one cell in the cube, without changing the network structure. For roll-up in topological OLAP, networks are not grouped but the reorganization is inside individual snapshots and a new generalized graph is built with a new topological structure. Lastly, a traditional data warehouse does not consider relationships between records.

Table 1. Comparison between traditional OLAP and Graph OLAP

	Traditional OLAP	Graph OLAP
Input	Facts in cuboids	A given network with snapshots
Output	Aggregated measures	A new network more generalized
Dimensions	Attributes	Informational and topological
Hierarchies	Yes	Yes (both for info. and topo. dimensions)
Measures	Numeric indicators	Aggregated graph measure
	Aggregation function (count, sum, average)	Measures coming from graph theory Specific aggregation functions
Operations	Roll-up, drill-down, slice & dice, pivot	Operations within informational or topological OLAP
Problems	Not considering links among data records	How taking interactions among entities into account

4.2 Literature Review

In recent years, many researchers have been interested in OLAP on information networks. Wei proposed a concept of link OLAP based on link-oriented analysis [17]. It extended entity analysis to link analysis. However, he did not propose new models or operations. Tian *et al.* introduced an operation called *SNAP* [16]. It can produce a summary graph by grouping nodes. Moreover, users can control the different resolutions of summaries by a k-SNAP operation. Chen *et al.* and Qu *et al.* proposed a data cube on graphs [3,15]. Chen's framework used the top-10 central method for visualization over the data cube. While Qu's proposal efficiently computed measures and user's requests with two measure properties: T-Distributiveness and T- Monotonicity. Zhao *et al.* introduced a new data warehouse model [19]. This model is called *Graph cube* and it supports a new class of user queries called *crossboid*. Their model considered network aggregation both on entities and relationships. Kampgem *et al.* presented a mapping

from linked data to data cube [8]. They integrated statistic linked data into format for loading into OLAP systems. Trifonova *et al.* presented an application for analyzing bibliographic networks [13] and they used star schema to design the data warehouse. It allowed data extraction from data cubes and authors using tabular and graphical views. Yin *et al.* defined a concept of entity dimensions to support two dimensions of heterogeneous networks [18]. They proposed *HMGraph OLAP* a data warehouse model using constellation schema. They designed novel operations named *Rotate* and *Stretch*. The previous studies did not mention how to improve performances on cube materialization. Therefore, Yin's approach demonstrated the strategy of index graphs. Researches are interested in studies about OLAP on information networks based on multidimensional and multilevel concepts. All of them have not provided a query language to support n-dimensional computations on graph OLAP. So Beheshti *et al.* proposed a graph data model and a query language extended SPARQL [1]. Their model considered both objects and links among them and there are two kinds of dimensions of information networks.

5 Proposed Framework

The previous works have been interested in the effectiveness and efficiency of Graph OLAP to provide OLAP on information networks. The major limitation of these studies is that building a data warehouse is limited to only one network.

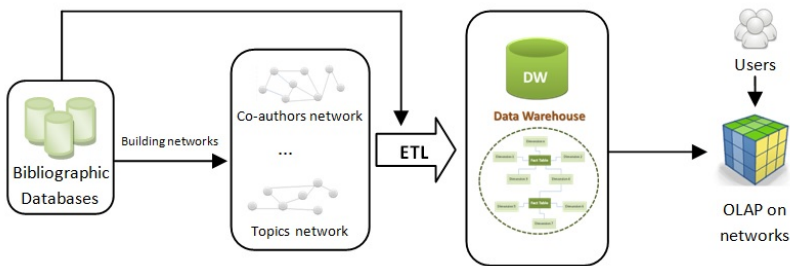


Fig. 2. The proposed framework

Our proposal is to design a data warehouse and OLAP analysis for several networks. The proposed framework is shown in figure 2. The starting point is databases of bibliographic data. We examine three online databases in computer science domain (DBLP, ACM and PASCAL) that allow us to collect publications under the form of XML data. Our idea is to build different networks from such databases: co-authors network, citations network, topics network, conferences network and so on. The networks are represented under the form of graphs. We would represent different actors and types of links as follows:

- co-author network is created with authors as vertices and co-author relationships as edges,
- in citation network, vertices are papers and edges represent a relationship between the cited and citing documents,
- topic network contains topic areas as vertices and the same area as relation,
- conference network contains names of conference as vertices and the same area as relation.

Many techniques can be used to extract knowledge from those networks such as data mining and SNA methods. The extracted knowledge can enrich networks. For example, clustering is useful to discover communities in many systems. It is to classify groups of entities that share similar properties and to provide the changes in the objects over time such as discovering organizational relations and identifying researcher communities. For instance, if we consider communities detection, the result could be used to enrich hierarchies with new levels of data. Ranking is to evaluate objects of networks based on mathematical or statistical functions. It needs to calculate the distance between objects and the cluster center. However, combining both clustering and ranking may lead to more better results. For instance, ranking authors related to conference cluster by using the number of citations, the most popular topics in each institutions and top-10 of researchers in research areas. SNA methods are used to study the relationships, analyze citations, compute communications and calculate indicators. For example, a concept of closeness is calculated as a relevant score for finding collaborators on similar topics. Degree centrality provides an answer to the question “who are the leaders among researchers or popular research topics?”.

Next, networks are loaded into a data warehouse through ETL process (Extract, Transform and Load). Different models are used to represent different networks. The structure of a data warehouse should be designed, it is based on the multidimensional model. It should be able to store the different networks and the related extracted knowledge. The fact can be a single node (an author, an institution) or a network (co-publications network). In our knowledge, there are many types of measure. Firstly, the measure can be classical like a numerical feature such as the number of papers, the number of citations and the number of downloads. The measure can be textual such as keywords. In social network analysis, the measure can be the centrality, the diameter or the similarity. Lastly, a network can be a measure in *Graph OLAP*. In term of the aggregation function, it depends on the types of measure and on hierarchy concept. With classical measures, the *SUM* or *AVERAGE* functions are well suitable, they can construct a group of authors by laboratory or institution. An other example in *Graph OLAP*, graph summarization is to cluster authors by relationships. In our framework, we want to have the several types of measure and the adapted aggregation functions.

At the end, we plan to create OLAP tools for network aggregation, visualization and navigation. We have to answer users’ queries such as navigating within other authors in collaboration who work on the same research topics. For efficient visualization and for network aggregation, we want to take into account

both attributes of nodes and links between nodes. More, we have to analyze the dynamic of a network (authors, publications) over time such as the most popular topics in each year. In order to create these new OLAP tools, we plan to combine data mining methods and OLAP operators.

Considering this proposed framework, we have identified several challenges. First, we have to build the several networks by extracting them from databases and we have to extract knowledge from networks in order to enrich them. For this double task, we have to consider the existing algorithms and data mining techniques would be very useful. Secondly, a big challenge is how to design the model for storing multi-networks and knowledge. We think that classical models cannot meet our needs and we probably led to invent a new model. Thirdly, we have to consider the ETL step. How to consider this phase both for networks and knowledge ? Last, there is a crucial challenge to provide analysis tools, dealing with the various considered networks. Innovative tools should be developed for users.

6 Conclusion

In this paper, we discussed the interest of combining OLAP technology and information networks in the context of bibliographic data analysis.

We presented a related work on the use of these two domains to emphasize how it is possible to combine them. We also proposed a tentative framework to analyze bibliographic data taking benefit from these two areas and we addressed the main challenges to solve. The main ideas are (i) building various networks from the bibliographic databases such as DBLP, ACM... (co-author network, citations network, topics network, conferences network) ; (ii) building a data warehouse with the appropriate model to explore these information ; (iii) applying data mining techniques to enrich this information (such as detecting communities to enrich dimension hierarchies of the data warehouse) ; and (iv) developing appropriate tools (inspired from OLAP navigation process) for visualizing these data. Various problems have to be solved, such as summarizability and topological issues. From a technical point of view, we need to explore existing tools and their usage (for instance graph database tool such as neo4j¹).

In terms of perspectives of this preliminary work, we aim at dealing with every underlined challenges to provide a complete solution implementing our framework that combines OLAP technology, data mining and information networks to deal with bibliographic data.

References

1. Beheshti, S.-M.-R., Benatallah, B., Motahari-Nezhad, H.R., Allahbakhsh, M.: A Framework and a Language for On-Line Analytical Processing on Graphs. In: Wang, X.S., Cruz, I., Delis, A., Huang, G. (eds.) WISE 2012. LNCS, vol. 7651, pp. 213–227. Springer, Heidelberg (2012)

¹ www.neo4j.org

2. Chaudhuri, S., Dayal, U.: An Overview of Data Warehousing and OLAP Technology. *ACM SIGMOD* 26(1), 65–74 (1997)
3. Chen, C., Yan, X., Zhu, F., Han, J., Yu, P.S.: Graph OLAP: Towards online analytical processing on graphs. In: *ICDM 2008*, pp. 103–112 (2008)
4. Deng, H., King, I., Lyu, M.R.: Formal Models for Expert Finding on DBLP Bibliography Data. In: *ICDM 2008*, pp. 163–172 (2008)
5. Gupta, M., Aggarwal, C.C., Han, J., Sun, Y.: Evolutionary Clustering and Analysis of Bibliographic Networks. In: *ASONAM 2011*, pp. 63–70 (2011)
6. Han, J.: Mining Heterogeneous Information Networks by Exploring the Power of Links. In: Gama, J., Costa, V.S., Jorge, A.M., Brazdil, P.B. (eds.) *DS 2009*. LNCS, vol. 5808, pp. 13–30. Springer, Heidelberg (2009)
7. Huang, Z., Yan, Y., Qiu, Y., Qiao, S.: Exploring Emergent Semantic Communities from DBLP Bibliography Database. In: *ASONAM 2009*, pp. 219–214 (2009)
8. Kampgen, B., Harth, A.: Transforming statistical linked data for use in OLAP systems. In: *I-SEMANTICS*, pp. 33–40 (2011)
9. Klink, S., Reuther, P., Weber, A., Walter, B., Ley, M.: Analysing Social Networks Within Bibliographical Data. In: Bressan, S., Küng, J., Wagner, R. (eds.) *DEXA 2006*. LNCS, vol. 4080, pp. 234–243. Springer, Heidelberg (2006)
10. Muhlenbach, F., Lallich, S.: Discovering Research Communities by Clustering Bibliographical Data. In: *WI-IAT 2010*, vol. 1, pp. 500–507 (2009)
11. Pham, M.C., Klamma, R.: The Structure of the Computer Science Knowledge Network. In: *ASONAM 2010*, pp. 17–24 (2010)
12. Sun, Y., Han, J., Zhao, P., Yin, Z., Cheng, H., Wu, T.: RankClus: integrating clustering with ranking for heterogeneous information network analysis. In: *EDBT 2009*, pp. 565–576 (2009)
13. Trifonova, T.G.: Warehousing and OLAP Analysis of Bibliographic Data. *Intelligent Information Management* 3, 109–197 (2011)
14. Yu, P.S.: Information networks mining and analysis. In: Du, X., Fan, W., Wang, J., Peng, Z., Sharaf, M.A. (eds.) *APWeb 2011*. LNCS, vol. 6612, pp. 1–2. Springer, Heidelberg (2011)
15. Qu, Q., Zhu, F., Yan, X., Han, J., Yu, P.S., Li, H.: Efficient Topological OLAP on Information Networks. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) *DASFAA 2011, Part I*. LNCS, vol. 6587, pp. 389–403. Springer, Heidelberg (2011)
16. Tian, Y., Hankins, R.A., Patel, L.M.: Efficient Aggregation for Graph Summarization. In: *SIGMOD Conference*, pp. 567–580 (2008)
17. Wei, W.: Complex network virtualization and link OLAP (2007)
18. Yin, M., Wu, B., Aeng, Z.: HMGraph OLAP: a Novel Framework for Multi-dimensional Heterogeneous Network Analysis. In: *DOLAP 2012*, pp. 137–144 (2012)
19. Zhao, P., Li, X., Xin, D., Han, J.: Graph cube: on warehousing and OLAP multi-dimensional networks. In: *SIGMOD 2011*, pp. 853–864 (2011)

Part VI

Doctoral Consortium

Spatial Indexes for Simplicial and Cellular Meshes

Riccardo Fellegara

Department of Computer Science, Bioengineering, Robotics and System Engineering,
University of Genova, Italy
`riccardo.fellegara@unige.it`

Abstract. We address the problem of performing spatial and topological queries on simplicial and cellular meshes. These arise in several application domains including 3D GIS, scientific visualization and finite element analysis. Firstly, we present a family of spatial indexes for tetrahedral meshes, that we call tetrahedral trees. Then, we present the PR-star octree, that is a combined spatial data structure for performing efficient topological queries on simplicial meshes. Finally, we propose to extend these frameworks to arbitrary dimensions and to larger class of meshes, such as non-simplicial meshes.

1 Introduction

The efficient representation of geometric shapes (solid objects, surfaces, terrains, etc.) is an active research topic in several fields, including geometric modeling, computer graphics, scientific visualization, and geographic data processing. Geometric shapes are usually discretized as meshes made of polyhedral cells, or of simplices. Triangular and tetrahedral meshes are Examples of simplicial meshes. Managing meshes in three dimensions (and higher) is not a simple task since the data structures proposed for three dimensional meshes are quite large and are even larger when considering meshes in four dimensions and higher. Current implementations cannot always keep large meshes entirely in system memory (*in-core*) and some *out-of-core* approaches, for example using a spatial data base, are intrinsically slow.

There are two fundamental categories of queries proposed in the literature for interacting with simplicial meshes: those based on spatial locality and those based on topological connectivity. Spatial information can be retrieved through *spatial queries*, such as point location queries, box queries (i.e., finding the simplices of the mesh inside a rectangle or a parallelepiped) and ray intersection queries (i.e, finding the intersection of a ray with the cells of the mesh). Spatial queries are required to understand the relation between the cells of the mesh and arbitrary regions in space, and are thus based on geometric information. Topological connectivity information can be retrieved through *topological queries*. Example of topological queries are, for instance in a triangle mesh, finding the triangles adjacent to a given triangle through an edge or a vertex; or finding

all triangles incident to a given vertex or edge. Thus, topological queries are necessary to perform navigation over the mesh.

In general, topological data structures [24,21] tend to be inefficient for spatial queries and spatial indexes exhibit a high overload when executing topological queries. The aim of this research is to propose and investigate new data structures for simplicial and cellular meshes in three dimensions and higher, based on spatial indexes and topological data structures. The final goal is to find optimized data structures that combine the features of topological and spatial approach in a common framework.

We have investigated and studied two research problems in geometric modeling: (i) the design and the implementation of a framework for efficiently managing spatial indexes for tetrahedral meshes; (ii) the design and the implementation of a framework based on topological spatial indexes, which could efficiently compute topological relations, and can efficiently execute different tasks, such as mesh simplification and curvature computation.

The remainder of this paper is organized as follows. In Section 2, we present the state of the art on spatial indexes. In Section 3 we present the framework for efficiently managing spatial indexes for tetrahedral meshes. In Section 4 we present the framework based on topological spatial indexes, and in Section 4.1, an application of such framework, that extracts efficiently *morphological features*. Finally, in Section 5 we describe some possible future developments of the presented frameworks.

2 State of the Art

A hierarchical spatial index is a data structure used for indexing spatial information, such as points, polygonal maps and objects in the Euclidean space, and subdivides the domain, accordingly to the distribution of the data. In literature, two sub-families of these hierarchical spatial indexes have been defined: those that index the complexes following an *object-based* decomposition, such as *R-trees* [13], and those that index the complexes following a *space-based* decomposition.

Hierarchical spatial indexes, based on the space decomposition, apply a nested refinement of the complex domain, covering its domain. This decomposition is usually represented in the form of a tree, that defines a hierarchical relationship among the set of nodes in the tree, where a parent node's children are generated during the refinement. The root of a tree covers the entire domain and is the only node without a parent. Nodes with children are referred to as internal nodes of the tree, while those without children are referred to as leaf nodes of the tree.

There are two main families of these hierarchical data structure: one based on the quaternary/octal tree and one based on binary tree, i.e. *kD-trees*.

In literature, two approaches are available to represent spatial information into a tree: inside the internal nodes or inside the leaf nodes. The trees that store the spatial information into the internal nodes, are dependent on the insertion order and intrinsically static structures. On the contrary, those trees that store

the spatial information into the leaves are order independent and, thus, allow the update/removal of indexed data without modify the entire tree shape.

Point Region (PR)-quadtree and *2D PR-kD-trees* [22,29] are hierarchical data structures that represent point sets. In such indexes, a block is recursively subdivided if it contains a number of points higher than a given capacity threshold. The shape of the tree is independent of the order in which the points are inserted, since the subdivision is based on the domain and on the data points, and the points are only in the leaf blocks. In three dimensions the direct extension of these index are *PR-octrees* and *3D PR-kD-trees*, that index point sets in the 3D Euclidean space.

The class of *Polygonal Map* (PM)-quadtrees [30] extends the PR-quadtree to represent polygonal maps in 2D. All the indexes of this class maintain a list of edges in the leaf blocks and are based on different subdivision rules for the map.

The *randomized Polygonal Map* (PMR) quadtree [19] is a spatial index for a collection of edges in the plane, not necessarily forming a polygonal map and can be used for spatial objects in the plane [14]. The PMR quadtree uses a user-determined splitting threshold. If the insertion of an edge causes the number of edges in a leaf block to exceed the splitting threshold, the block is split only once at this time. The rationale for this choice is that it avoids excessive splitting. This gives rise to a probabilistic behavior in the sense that the order in which the segments are inserted affects the shape of the resulting tree. In [16] it has been proven that in a PMR-quadtree the number of nodes is proportional to the number of line segments and is independent of the maximum depth of the tree.

The PM index family has been also extended to encode polyhedral objects [3,18,29], where octrees have been used to index the surface bounding a polyhedral object in space, and thus, are called *PM-octrees*. Leaves are of four types: *empty* (no intersection with the boundary surface), *vertex* (all intersecting elements incident into an internal vertex), *edge* (all intersecting elements incident into a crossing edge), or *face* (only one crossing face). The configuration inside each leaf block can be maintained in different ways. Carlbon et al. [3] explicitly store the boundary elements present in the block, while Navazo [18] stores just the equations.

Space Partition (SP)-octrees [2] maintain information about the elements of the boundary surface of an object not only in leaves, but also in internal nodes. Thus, internal nodes give an approximate description of the object boundary. The subdivision stops when the object portion lying in the block can be defined as the intersection of the planes of faces intersecting the block (i.e., it is locally convex or locally concave).

Spatial indexes have been recently proposed for triangle and tetrahedral meshes. In [25] an efficient indexing techniques for generic spatial queries on tetrahedral meshes is presented which works *out-of-core*. A query processing technique for spatial queries, called *Directed Local Search* (DLS), is presented and takes advantage of the mesh connectivity and its efficiency is independent of the complexity of the mesh geometry. DLS can be easily implemented in a database system without requiring the development of new access methods, but has all the limitations of the topology-based approaches.

The PM_2 -quadtree has been extended to index triangle meshes giving rise to the *PM₂-Triangle quadtree* [5]. This index is designed for performing spatial queries on triangle-based terrain models. In such an index, a node use a compact encoding to index the geometric entities that intersect the leaf, that significantly reduce the overall storage requirements of the index.

In literature the representation of these meshes, usually, is guided by topological data structures [24,21,12,1], that are data structures that support the efficient reconstruction of a subset of the local topological connectivity of the mesh as well as navigation. The topological connectivity is identified through some relations, called *topological relations*, between the simplices that form the complexes. These data structures are efficient to answer queries that require an intense navigation of the object, while are limited at answering those queries that require a spatial navigation, such as point location and box queries. To enable these queries, it is needed an approach that is based on the notion of walk [9,17,10,4], that, thought, cannot (fully) answer these queries if into the object domain there are concavities or "holes".

Pointer-less representations have been developed in the literature mainly for different variants of quadtrees, and octrees (see, for instance, the linear quadtrees [11]). The pointer-based approach is not well suited for encoding large indexes (especially octrees) and for implementing disk-based structures. Pointer-less representations only store the leaf blocks in the tree, which are defined and localized through a *location code*, where the location of a leaf is defined by a sequence of bits denoting the corresponding root-to-leaf path. A PMR-quadtree implementation using location codes is presented in [14].

3 Tetrahedral Trees

We have first developed a family of spatial indexes for tetrahedralized shapes, that we called *tetrahedral trees* [7], built over a spatial index, an *octree* or a *kD-tree*, that associate portions of the discretized shape to the leaf nodes and are organized according to a specific subdivision rule.

The simplest rule gives rise to the *PR-Tetrahedral (PR-T) Tree*, which is based only on the vertices of the mesh and thus leads to *PR-octree* or to a *PR-kD-Tree* [29]. The second rule provides *Centroid-Tetrahedral (C-T) Tree*, which is still point-based, but uses the centroids of the tetrahedra as points. The third rule leads to the *PMR-Tetrahedral (PMR-T) Tree*, which adopts the basic principle used in the PMR-quadtree [19]: it subdivides a block when the number of tetrahedra intersected by the block is greater than a certain threshold, but the subdivision is done only once at each insertion. The fourth rule generates the *PM-Tetrahedral (PM-T) Tree*, which subdivides a block based on the number of vertices and on the topology of the tetrahedral mesh in that node. By combining the two spatial data structures (octree and kD-tree) and the four subdivision criteria, we obtain eight different spatial indexes. A 2D example illustrating how the space is subdivided, changing the rule and/or the spatial index, is shown in Figure 1.

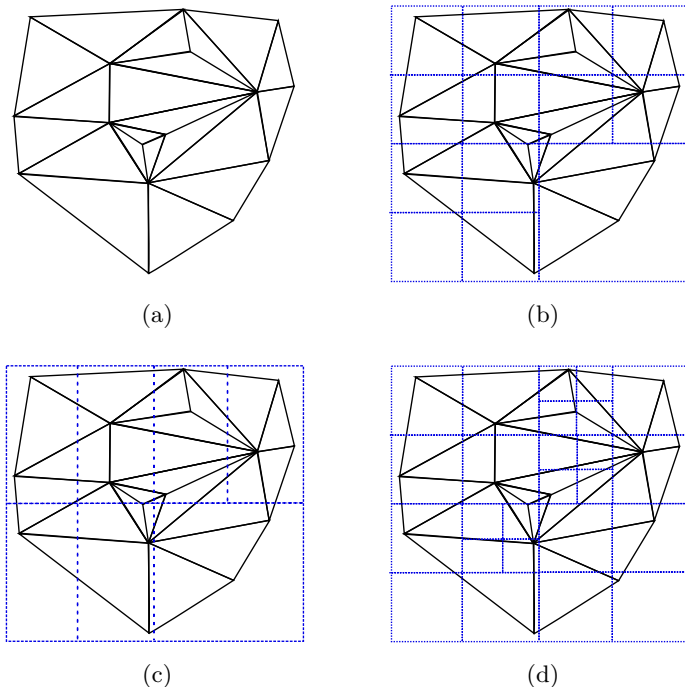


Fig. 1. Example of spatial index subdivision for a triangulation. (a) shows the original mesh, (b) shows the subdivision obtained by a PR-T-quadtrees index, with threshold equal to two vertex for leaf, (c) shows the subdivision obtained by a PR-T-kD-trees index, with the same threshold as (a), and (d) shows the subdivision obtained by a PM-T-quadtrees index, with the same threshold for the vertices, and with threshold 5 for the triangles.

We have considered two basic queries on tetrahedral meshes, which are common to most applications, to evaluate our spatial indexes. These are the *point location query* and the *window query*. Our results show that the *PMR-T tree* has, in general, a better performance in queries, and it also has a moderate memory overhead. Only the *C-T tree* is more compact than the *PMR-T tree*, but queries are less efficient. The other two indexes have noticeably larger size and construction times, with just a slight improvement in query times, especially for larger meshes. This result matches the one reported in [28], which compares *PMR-quadtrees* and *PM-quadtrees* used to index the set of segments forming a polygonal map in geographic applications. Using *kD-trees*, instead of *octrees*, provides slightly larger memory requirements and no gain in query times, which are even worse for queries involving small portions of the domain. More recent research has enhanced the tetrahedral trees framework by adding a new space criterion index and a thorough comparison with topological state-of-the-art data structures [8].

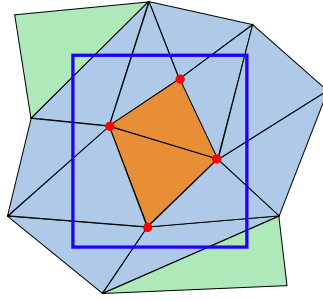


Fig. 2. A leaf node in a PR-star octree (shown in 2D with bucket threshold $k_v = 4$) encodes a set of vertices and all tetrahedra incident in those vertices. In orange the tetrahedra that are completely indexed by the leaf (blue rectangle), in blue those that are incident into a vertex contained by the leaf, and in green those that geometrically intersect the leaf but are not incident into one of the internal vertices, and thus are not indexed by the leaf.

4 The PR-Star Octree

We have designed and implemented a framework based on topological spatial indexes that leads to a new data structure, the *PR-star octree* [31], in which we obtain local topological connectivity of a tetrahedral mesh through its spatial locality. In contrast to topological data structures, which have focused on the adjacencies or incidences of the mesh elements, we use a spatial data structure on its embedding space to locally reconstruct the optimal application-dependent topological representation at runtime using the sorted geometry available from our spatial index. Thus, the innovative feature of our approach is in computing topology through space: local spatial sorting allows the efficient reconstruction of the local mesh connectivity. Although this increases the cost of a single operation due to the construction of the local data structure, this cost is amortized over multiple accesses to elements within the same region. Moreover, by recovering the memory associated with each local data structure after the processing of that part of the mesh has completed, we achieve significant memory savings with respect to global topological data structures.

The PR-star octree combines the indexed tetrahedral mesh representation with an augmented PR octree that also indexes the set of tetrahedra in the star of its indexed vertices, as shown in Figure 2. Thus, a PR-star octree over a tetrahedral mesh is represented using an array of vertices V , encoding the geometry of the mesh, an array of tetrahedra T and an augmented PR octree N , whose leaf nodes index the set of vertices within its domain, as well as the set of all tetrahedra incident in these vertices. Then, we exploit the spatial locality of N , by re-indexing the vertex array V and the tetrahedra array T . After this stage, the leaf nodes of N index a contiguous range of vertices from V , the internal nodes of N index a contiguous range of vertices from V indexed by its descendants. Finally, we have shown that the PR-star tree is also a dynamic data

structure, which adapts to the changes in the underlying mesh, by implementing to this framework the simplification of the underlying mesh obtained through iterative edge-collapse.

4.1 Morphological Feature Extraction on the PR-Star Octree

As an application of PR-star, we have designed and implemented a framework [6] which can efficiently compute and extract morphological features in 2D and 3D scalar fields, such as critical points and their regions of influence, based on discrete Morse theory. To compute the discrete gradient vector we have adapted the algorithm of Robins et al. [27] for extracting the discrete Morse gradient field to simplicial meshes. To extract morphological features using the above gradient vector we have defined a streaming algorithm, that extracts all such features in a single pass through the tree leaf nodes. Figure 3 shows two examples of extracted features. We do not assume the existence of a global gradient vector field, while we only generate a local vector gradient field into the leaves of a PR-star tree. Our results show that the storage required by the data structures for encoding the mesh and by the auxiliary representations, used during the computation of the gradient and for the extraction of the feature, PR-octree implementation uses from 30% to 40% less memory than a state of the art topological connectivity data structure [24]. Considering the relative performances across the whole feature extraction process (gradient computation plus feature extraction), the PR-star is more efficient from 5% to 30%.

Then, we have extended this framework in [32] by developing a new encoding for discrete vector fields, that is compact and suitable for combination with any topological data structure encoding just the vertices and tetrahedra of the mesh. We use a duality argument to define the cells of the descending Morse complex in terms of the supplied (primal) tetrahedral mesh and those of the ascending complex in terms of its dual mesh. The Morse-Smale complex has been then described combinatorially as collections of cells from the intersection of the primal and dual meshes. This leads to simple descriptions of morphological features in terms of only the vertices and tetrahedra of the primal mesh. The compact encoding of discrete vector fields uses the *local frame* representation, which associates information with the tetrahedra in the primal mesh, and which we apply to the discrete Morse gradient field. Our results show that the PR-star octree implementation uses from 20% to 35% less memory than the most common state-of-the-art topological data structure [24], at the expense of some additional computation, for smaller mesh is up to two times less efficient, while for bigger meshes obtain almost the same timings.

5 Ongoing Research

All the research done so far has been concentrated to obtain an efficient management of manifold triangular and tetrahedral meshes, embedded in three-dimensional Euclidean space, and we have developed the tetrahedral trees and the PR-star tree.

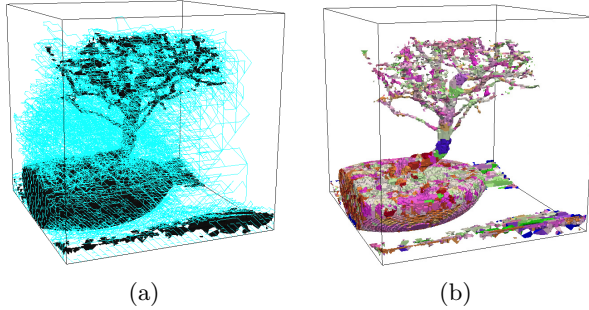


Fig. 3. Example of morphological feature extraction on a tetrahedral dataset. (a) shows the extracted 1-descending manifolds, while (b) shows the extracted 3-descending manifolds.

Thanks to the intrinsic dimension-independent feature of the PR-star tree and to fact that the representation of the PR-star tree is independent of the indexed data, we plan to extend the PR-star tree and its implementation to higher dimensions and to larger class of meshes, such as non-simplicial meshes.

Firstly, we plan to extend our implementation to handle general complexes, such as non-manifold shapes. Non-manifold shapes are shapes in the Euclidean space for which the neighborhood of each of its points is not homeomorphic to an open ball, or to an open half-ball. These objects arise in several applications, for example in the *idealization* process (the process that idealize a real complex object) for preparing an object for finite element simulations.

Our next stage is the handling of non-simplicial meshes. These meshes are widely used in geometry processing (quad meshes), finite element analysis (FEM) and in CAD and CAM systems. In three dimensions these meshes are usually quad-meshes, arbitrary polygonal meshes and unstructured hexahedral meshes.

We are not limiting the dimension of these meshes, both simplicial and non-simplicial, and manifold and non-manifold, as we want to proceed to higher dimensions, to stress our implementation, for both static applications, such as feature extraction queries, and dynamic applications, such as mesh simplification.

We want to apply the higher dimensional PR-star tree to hyper meshes (tetrahedral meshes) in 4D space which represent iso-surfaces of time-varying scalar fields or sequences of mesh animation. Displaying iso-surfaces is still one of the most commonly used techniques to analyze three dimensional scalar fields, and the efficient computation and rendering of iso-surfaces plays a crucial role [23,15,20,26]. This "dynamic geometry" can be represented as a set of iso-surfaces that are extracted individually at certain time steps, or by representing the whole sequence as a four-dimensional tetrahedral mesh. The iso-surface at a specific time step can then be computed by intersecting the tetrahedral mesh with a three-dimensional hyperplane.

Acknowledgements. Many thanks to Leila De Floriani and Kenneth Weiss for their many helpful comments and suggestions. This work has been partially supported by the Italian Ministry of Education and Research under the PRIN 2009 program, and by the National Science Foundation under grant number IIS-1116747.

References

1. Canino, D., De Floriani, L., Weiss, K.: IA*: An adjacency-based representation for non-manifold simplicial shapes in arbitrary dimensions. *Computers & Graphics* 35(3), 747–753 (2011)
2. Cano, P., Torres, J.: Representation of polyhedral objects using SP-octrees. *Journal of WSCG* 10(1), 95–101 (2002)
3. Carlbom, I., Chakravarty, I., Vanderschel, D.: A hierarchical data structure for representing the spatial decomposition of 3-D objects. *IEEE Computer Graphics and Applications* 5(4), 24–31 (1985)
4. De Carufel, J., Dillabaugh, C., Maheshwari, A.: Point location in well-shaped meshes using jump-and-walk. In: *Canadian Conference on Computational Geometry (CCCG)*, pp. 147–152 (2011)
5. De Floriani, L., Facinoli, M., Magillo, P., Dimitri, D.: A hierarchical spatial index for triangulated surfaces. In: *Proceedings of the Third International Conference on Computer Graphics Theory and Applications (GRAPP 2008)*, pp. 86–91 (2008)
6. De Floriani, L., Fellegara, R., Iuricich, F., Weiss, K.: A spatial approach to morphological feature extraction from irregularly sampled scalar fields. In: *Proceedings of the Third ACM SIGSPATIAL International Workshop on GeoStreaming*, pp. 40–47. ACM (2012)
7. De Floriani, L., Fellegara, R., Magillo, P.: Spatial Indexing on Tetrahedral Meshes. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 506–509. ACM (2010)
8. De Floriani, L., Fellegara, R., Magillo, P., Weiss, K.: Tetrahedral trees: A family of hierarchical spatial indexes for tetrahedral meshes (in Preparation)
9. Devillers, O., Pion, S., Teillaud, M.: Walking in a triangulation. In: *Proceedings of the Seventeenth Annual Symposium on Computational Geometry*, pp. 106–114. ACM (2001)
10. Dillabaugh, C.: I/O efficient path traversal in well-shaped tetrahedral meshes (2010)
11. Gargantini, I.: An effective way to represent quadtrees. *Communications of the ACM* 25(12), 905–910 (1982)
12. Gurung, T., Rossignac, J.: SOT: A compact representation for tetrahedral meshes. In: *Proceedings SIAM/ACM Geometric and Physical Modeling, SPM 2010, San Francisco, USA*, pp. 79–88 (2009)
13. Guttman, A.: R-trees: a dynamic index structure for spatial searching, vol. 1. ACM (1984)
14. Hjalton, G., Samet, H.: Speeding up construction of PMR quadtree-based spatial indexes. *The VLDB Journal — The International Journal on Very Large Data Bases* 11(2), 137 (2002)
15. Houston, B., Nielsen, M.B., Batty, C., Nilsson, O., Museth, K.: Hierarchical rle level set: A compact and versatile deformable surface representation. *ACM Transactions on Graphics (TOG)* 25(1), 151–175 (2006)

16. Lindenbaum, M., Samet, H., Hjaltason, G.R.: A probabilistic analysis of trie-based sorting of large collections of line segments in spatial databases. *SIAM Journal on Computing* 35(1), 22–58 (2005)
17. Mücke, E., Saias, I., Zhu, B.: Fast randomized point location without preprocessing in two- and three-dimensional delaunay triangulations. In: *Proceedings of the Twelfth Annual Symposium on Computational Geometry*, pp. 274–283. ACM (1996)
18. Navazo, I.: Extended octree representation of general solids with plane faces: model structure and algorithms. *Computer & Graphics* 13(1), 5–16 (1989)
19. Nelson, R., Samet, H.: A population analysis for hierarchical data structures. In: *Proc. ACM SIGMOD Conference, San Francisco, CA, USA*, pp. 270–277 (1987)
20. Nielsen, M.B., Museth, K.: Dynamic tubular grid: An efficient data structure and algorithms for high resolution level sets. *Journal of Scientific Computing* 26(3), 261–299 (2006)
21. Nielson, G.M.: Tools for triangulations and tetrahedralizations and constructing functions defined over them. In: Nielson, G.M., Hagen, H., Müller, H. (eds.) *Scientific Visualization: Overviews, Methodologies and Techniques*, vol. ch. 20, pp. 429–525. IEEE Computer Society, Silver Spring (1997)
22. Orenstein, J.: Multidimensional tries used for associative searching. *INFO. PROC. LETT.* 14(4), 150–157 (1982)
23. Osher, S., Fedkiw, R.: *Level set methods and dynamic implicit surfaces*, vol. 153. Springer (2003)
24. Paoluzzi, A., Bernardini, F., Cattani, C., Ferrucci, V.: Dimension-independent modeling with simplicial complexes. *ACM Transactions on Graphics (TOG)* 12(1), 56–102 (1993)
25. Papadomanolakis, S., Ailamaki, A., Lopez, J.C., Tu, T., O’Hallaron, D.R., Heber, G.: Efficient query processing on unstructured tetrahedral meshes. In: *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Datsa*, pp. 551–562. ACM (2006)
26. Ponchio, F., Hormann, K.: Interactive rendering of dynamic geometry. *Visualization and Computer Graphics, IEEE Transactions on* 14(4), 914–925 (2008)
27. Robins, V., Wood, P.J., Sheppard, A.P.: Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE Trans. Pattern Anal. Mach. Intell.* 33(8), 1646–1658 (2011)
28. Samet, H.: *The Design and analysis of spatial data structure*. Addison-Wesley, Reading (1990)
29. Samet, H.: *Foundations of multidimensional and metric data structures*. Morgan Kaufmann (2006)
30. Samet, H., Webber, R.: Storing a collection of polygons using quadtrees. *ACM Transactions on Graphics (TOG)* 4(3), 182–222 (1985)
31. Weiss, K., Fellegara, R., De Floriani, L., Velloso, M.: The PR-star octree: A spatio-topological data structure for tetrahedral meshes. In: *Proceedings ACM SIGSPATIAL GIS, GIS 2011*. ACM (November 2011)
32. Weiss, K., Iuricich, F., Fellegara, R., Floriani, L.D.: A primal/dual representation for discrete morse complexes on tetrahedral meshes. In: *Computer Graphics Forum (CGF)* (to appear, 2013), also presented at 15th EuroVis Eurographics/IEEE Symposium on Visualization

Mathematical Methods of Tensor Factorization Applied to Recommender Systems

Giuseppe Ricci, Marco de Gemmis, and Giovanni Semeraro

Department of Computer Science University of Bari Aldo Moro, Italy
giuseppe.ricci@uniba.it, {marco.degemmis,giovanni.semeraro}@uniba.it
<http://www.di.uniba.it/~swap/index.php>

Abstract. On internet today, an overabundance of information can be accessed, making it difficult for users to process and evaluate options and make appropriate choices. This phenomenon is known as *information overload*. Over time, various methods of information filtering have been introduced in order to assist users in choosing what may be of their interest. Recommender Systems (RS) [14] are techniques for information filtering which play an important role in e-commerce, advertising, e-mail filtering, etc. Therefore, RS are an answer, though partial, to the problem of information overload. Recommendation algorithms need to be continuously updated because of a constant increase in both the quantity of information and ways of access to that information, which define the different contexts of information use. The research of more effective and more efficient methods than those currently known in literature is also stimulated by the interests of industrial research in this field, as demonstrated by the Netflix Prize Contest, the open competition for the best algorithm to predict user ratings for films, based on previous ratings. The contest showed the superiority of mathematical methods that discover latent factors which drives user-item similarity, with respect to classical collaborative filtering algorithms. With the ever-increasing information available in digital archives and textual databases, the challenge of implementing personalized filters has become the challenge of designing algorithms able to manage huge amounts of data for the elicitation of user needs and preferences. In recent years, matrix factorization techniques have proved to be a quite promising solution to the problem of designing efficient filtering algorithms in the *Big Data* Era. The main contribution of this paper is an analysis of these methods, which focuses on tensor factorization techniques, as well as the definition of a method for tensor factorization suitable for recommender systems.

Keywords: Recommender Systems, Matrix Factorization, Tensor Factorization, PARAFAC/CANDECOMP.

1 Matrix Factorization

Recommender systems guide users in a personalized way to interesting or useful objects in a large space of possible options, by providing a list of suggested

items that fits their interests. For example, Netflix, a provider of on-demand Internet streaming video and flat rate DVD-by-mail in the United States, adopts a recommendation algorithm to predict user interests for films, based on feedback provided by users on previously watched items. The most widely adopted recommendation techniques in literature are content-based and collaborative filtering ones.

Matrix Factorization (MF) techniques fall in the class of collaborative filtering (CF) methods and, particularly, in the class of latent factor models [10], which assume that similarity between users and items is induced by some factors hidden in the data. These models attempt to explain the ratings by characterizing both items and users with the objective of disclosing the latent features deduced from ratings. In the same way a person can naturally define the characteristics of a movie (such as genre, key players, duration, etc.), methods based on latent factors infer this characteristic data without exactly knowing each feature. In this case, latent factor models build a matrix of users and items (movies) and each element is associated with a vector of characteristics. MF techniques represent users and items by vectors of features derived from ratings given by users for the items seen or tried. A high correspondence between user and item factors leads to a recommendation. RS data are collected in a matrix called *user-item matrix*: rows are referred to users and columns to items; the intersection between one row and one column is the rating given by the user. Missing values correspond to movies not rated by the user.

Let U be the set of users, D the set of items, R the matrix of ratings. MF aims to factorize R into two matrices P and Q such that their product approximates R : $R \approx P \times Q^T$. Each row of P represents the strength of the association between user and k latent features. Similarly, each column of Q represents the strength of the association between an item and the latent features. Let p_i be the i -th row of P and q_j the j -th row of Q . They are the user profile vector and the item profile vector respectively, which represent the projection of user i and item j in a common space of k latent features. The scalar product $p_i \cdot q_j^T$ approximates the rating r_{ij} of user i for item j : $\hat{r}_{ij} = p_i \cdot q_j^T$. Once these vectors are discovered, recommendations are calculated using the expression of \hat{r}_{ij} . A factorization used in the literature is *Singular Value Decomposition* (SVD), introduced by Simon Funk in the NetFlix Prize [5], [3], has the objective of reducing the dimensionality, i.e. the rank, of the user-item matrix, in order to capture latent relationships between users and items [15]. Different SVD algorithms were used in RS literature: in [15], the authors uses a small SVD obtained retaining only $k \ll r$ singular values by discarding other entries; in [11], the authors propose an algorithm to perform SVD on large matrices, by focusing the study on parameters that affect the convergence speed; in [9], Koren presents an approach oriented on factor models which projected users and items in the same latent space where some measures for comparison are defined. He propose several versions of SVD with the objective of having better recommendations as well as good scalability.

2 Tensor Factorization

The main limitation of MF techniques is that they take into account only the standard profile of users and items. This does not allow to integrate further information such as context. For example, if a user watches a movie at home with his children, he will choose a movie whose genre is suitable for families. Indeed, in another context (friends or colleagues), the same user might prefer other kind of movies. Contextual information (the place where the user see the movie, the device, the company, etc.) cannot be managed with simple user-item matrices. *Tensors*, which can be seen as higher-dimensional arrays of numbers [8], might be exploited in order to include additional contextual information in the recommendation process [2]. In standard multivariate data analysis, data are arranged in a two-dimensional structure, but for a wide variety of domains, more appropriate structures are required for taking into account more dimensions. The techniques that generalize the MF factorization can also be applied to tensors. Two particular tensor decompositions [8] can be considered to be higher-order extensions of matrix singular value decomposition:

- **PARallel FACTor analysis** or CANonical DECOMPosition (PARAFAC/CANDECOMP) [4], [6], which decomposes a tensor as a sum of rank-one tensors;
- **High Order Singular Value Decomposition** (HOSVD) [12], which is a higher-order form of Principal Component Analysis (PCA).

In RS literature, the most frequently used technique for Tensor Factorization (TF) is HOSVD, which is a generalization of the SVD for matrices. This technique decomposes the initial tensor in N matrices (where N is the size of the tensor) and a tensor whose size is smaller than the original one.

HOSVD is used in [7], where the factorization of a tensor is applied to manage data for users, movies, user ratings and contextual information such as age, day of the week, companion. A third-order tensor is constructed and HOSVD is applied to factorize it into three matrices and one core tensor. Recommendation score for a single user i , item j and context k is computed by using these matrices and tensor. Another application of HOSVD for TF is described in [13], in the context of social tagging to predict a personalized list of tags for a user. Users' data, items and tags are stored in a third-order tensor which is factored by HOSVD, with the aim of discovering latent factors which bind the associations user-item, user-tag and tag-item. In [17], HOSVD is applied to the factorization of a tensor coming from a system of personalized web search, in order to discover the hidden relationships between objects typical of internet search: users, queries, web pages. Data related to user, query and web pages are collected in a third-order tensor that is decomposed with the technique of HOSVD.

The major advantage of HOSVD is the ability of simultaneously taking into account more dimensions. This allows for a better data modeling than standard SVD, since dimensionality reduction can be performed not only in one dimension but also separately for each dimension. But HOSVD is not an optimal tensor

decomposition, in the sense of least squares data fitting: the computation of HOSVD needs standard SVD computation only and has not the truncation property of the SVD, where truncating the first n singular values allows to find the best n -rank approximation of a given matrix. Despite this, the approximation obtained is not far from the optimal one and can be computed much faster. Since HOSVD cannot deal with missing values, they are treated as 0.

PARAFAC (PARallel FACTor analysis) is a decomposition method, which can be seen as a generalization of bilinear PCA. The PARAFAC model was independently proposed by Harshman [6] and by Carroll & Chang [4] who named the model *CANDECOMP* (CANonical DECOMPosition). A PARAFAC model of a three-dimensional array is given by three loading matrices A , B , and C with typical elements a_{if} , b_{jf} , and c_{kf} . The PARAFAC model is defined by the following structural model:

$$\hat{x}_{ijk} = \sum_{f=1}^F a_{if} b_{jf} c_{kf}. \quad (1)$$

where F is the number of rank-one components. PARAFAC is an alternative to HOSVD. One of the advantages of the PARAFAC is its simplicity which allows to use an analytical expression for solving the decomposition problem and to achieve linear scalability. Another advantage is linear computation time compared to HOSVD. PARAFAC does not collapse data, but it retains its natural three-dimensional structure. Despite PARAFAC mode's lack of orthogonality, Kruskal [8] showed that components are unique, up to permutation and scaling, under mild conditions.

In [16], PARAFAC is exploited for the computation of top- N context-aware recommendations of mobile applications. A tensor of three dimensions (users, items and context types) is factorized with PARAFAC. These dimensions are associated with the three factor matrices and used to calculate user preference for item i under context type k . In [1], PARAFAC is applied focusing on missing data. The authors developed a scalable algorithm called *CP-WOPT* (CP Weighted OPTimization), which uses first-order optimization to solve the weighted least squares objective function. Using extensive numerical experiments on simulated data sets, Acar et al. showed that CP-WOPT can successfully factor tensors with noise and up to 70% missing data. Moreover, CP-WOPT is significantly faster and accurate than the best published method in the literature [18].

3 CP-WOPT Adaptation: Preliminary Experiments

Our idea is to adapt CP-WOPT and to introduce it in the RS field, where the problem of missing values is very relevant, since the algorithm is suitable for very sparse user-items matrices. The adaptation allows the computation of a weighted factorization that models only known values, rather than simply employ 0 values for missing data. The main goal is to consider contextual information

about users and to apply the weighted PARAFAC decomposition to achieve precise recommendations. In order to reach this goal, we made a preliminary user study with 7 real users who were asked to rate a fixed number of movies (11) in the Movielens 100k dataset on the basis of three contextual factors: if they like to see the movie (i) at home or cinema; (ii) with friends or with partner; (iii) with or without family. Ratings range from 1 to 5 in the sense that, for each contextual comparison:

- rating 1 and 2 express a strong and a modest preference, respectively, for the first term;
- rating 3 expresses neutrality;
- rating 4 and 5 express a modest and a strong preference, respectively, for the second term.

Results are measured in terms of accuracy (acc), i.e. the percentage of known values correctly reconstructed and coverage (cov), i.e. the percentage of non-zero values returned. Under the assumption of 10^5 maximum iterations, we obtained $acc = 94.4\%$ and $cov = 91.7\%$. Although coming from a limited study, the values of these measures suggest we are moving in a correct direction and seem to promise encouraging results when applying the algorithm to more complex context-aware recommendation scenarios. Moreover, the experiment showed that it is possible to express, through the n-dimensional factorization, not only the recommendations for the single user, but also more specific suggestions about the consumption of an item. For instance, *American Pie* is typically watched at home, with friends and without family, while *Titanic* is preferably watched at cinema, with partner or family.

We performed also an in vitro preliminary experiment to test the adapted version of CP-WOPT on a subset of Movielens 100k dataset. We gave as input a tensor of dimensions 100 users, 150 movies, 21 occupations (the contextual factor) and we measured, besides acc and cov , also the classic Mean Average Error (MAE) and Root Mean Square Error (RMSE), in order to compare the results with those known in literature. The algorithm achieved: $acc = 92.09\%$, $cov = 99.96\%$, $MAE = 0.60$ and $RMSE = 0.93$, which are in line with results reported in literature.

In future we want to extend the evaluation of our version of CP-WOPT on tensor having high dimensionality extracted from Movielens dataset. In particular, we will investigate methods to assess whether contextual factors (occupation, company) influences the users' preferences, by using data mining techniques such as clustering. We plan also to test our approach in other domains such as news recommendation.

References

- [1] Acar, E., Dunlavy, D.M., Kolda, T.G., Mørup, M.: Scalable tensor factorizations with missing data. In: SDM 2010: Proceedings of the 2010 SIAM International Conference on Data Mining, Philadelphia, pp. 701–712. SIAM (April 2010)

- [2] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.* 23(1), 103–145 (2005)
- [3] Bennett, J., Lanning, S.: The netflix prize. In: *Proceedings of the KDD Cup Workshop 2007*, pp. 3–6. ACM, New York (2007)
- [4] Carroll, J., Chang, J.-J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika* 35(3), 283–319 (1970)
- [5] Funk, S.: Netflix update: Try this at home (2006), <http://sifter.org/~simon/journal/20061211.html>
- [6] Harshman, R.A.: Foundations of the PARAFAC Procedure: Models and Conditions for an "explanatory" Multi-modal Factor Analysis. In: *Working papers in phonetics*, vol. 1(16), University of California at Los Angeles (1970)
- [7] Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: *Proceedings of the fourth ACM Conference on Recommender Systems, RecSys 2010*, pp. 79–86. ACM, New York (2010)
- [8] Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Rev.* 51(3), 455–500 (2009)
- [9] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008*, pp. 426–434. ACM, New York (2008)
- [10] Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *IEEE Computer* 42(8), 30–37 (2009)
- [11] Kurucz, M., Benczúr, A.A., Torma, B.: Methods for large scale svd with missing values. In: *KDDCup 2007* (2007)
- [12] De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* 21, 1253–1278 (2000)
- [13] Rendle, S., Marinho, L.B., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: *KDD*, pp. 727–736 (2009)
- [14] Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): *Recommender Systems Handbook*. Springer (2011)
- [15] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: *Fifth International Conference on Computer and Information Science*, pp. 27–28 (2002)
- [16] Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Hanjalic, A., Oliver, N.: Tfmap: optimizing map for top-n context-aware recommendation. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2012*, pp. 155–164. ACM, New York (2012)
- [17] Sun, J.-T., Zeng, H.-J., Liu, H., Lu, Y., Chen, Z.: Cubesvd: a novel approach to personalized web search. In: *Proceedings of the 14th International Conference on World Wide Web, WWW 2005*, pp. 382–390. ACM, New York (2005)
- [18] Tomasi, G., Bro, R.: Parafac and missing values. *Chemometrics and Intelligent Laboratory Systems* 75(2), 163 (2005)

Extended Dynamic Weighted Majority Using Diversity to Handle Drifts

Parneeta Sidhu and M.P.S. Bhatia

Netaji Subhas Institute of Technology, University of Delhi, Dwarka, New Delhi, India
{parneeta07, bhatia.mps}@gmail.com

Abstract. Concept drift is the recent trend of online data. The distribution underlying the data is changing with time. There are many algorithms developed in the literature to handle such drifting data concepts. In our paper we are outlining the framework of our new approach to handle drifts which will be based on the concept of diversity. Diversity is the measure of variation in the predictive accuracy of ensemble members. Our approach would implement diversity concept first time on the *online approach that does not explicitly use a mechanism to handle drifts*. This type of online approach would give better accuracy at a slight increase in the running time and memory. In our paper we would also outline the main objectives behind our research and the state of the art in data stream mining.

Keywords: Data Streams, Concept Drift, Diversity, Ensemble Techniques.

1 Introduction

Online learning has been the new upcoming trend which has been very useful for applications where data is arriving continuously. This continuous flow of data forms a data stream which is dynamic in nature. We can have access to this data only “once” an arrival, after that the data is lost and new data arrives, which may have a different concept. This changing data concept is being widely used now-a-days in large number of applications like Market-Basket analysis [5], computer security, internet data, information filtering, credit fraud detection etc. In the next section, we will be discussing the objectives behind our research to work in this field of ensemble techniques of data stream mining. In the following section, we would discuss our main problem statement and discuss the state of the art to handle concept drift in data streams. In our paper, we would also give a brief information about the research results achieved so far and what would be the following phases we would pursue to achieve our goals.

2 Objectives

Concept drift is the main feature of data streams which is presently under study. Our area of work is also to develop a new methodology that would deeply study the

feature changes in data streams and give us the best accuracy. Here we are listing the main objectives that prompted me to do my doctorate under this field of concept drift in data stream mining.

- Knowledge extraction from the online data streams to predict the future trend.
- Study the dynamics in drifting data streams and increase the predictive accuracy.
- Explore the concept of diversity to develop the best approach.
- The new methodology should obey real time and space constraints.
- Our approach should be independent of any learning algorithm.
- It could be used as a wrapper over or be implemented within any learning algorithm.
- The approach could easily be extended to handle re-current or predictable drifts.

3 Background Knowledge

From the historical survey of machine learning literature, concept drifting algorithms have been broadly categorized into: Incremental approach and Online approach. In the Incremental approach, the data is processed as chunks of data of considerable large size and can be re-processed even using an offline learning algorithm. The latter approach, online learning algorithms, process each training instance once “on arrival” without the need for storage and reprocessing, and maintain a current hypothesis that reflects all the training instances so far. Further, the online approaches can be divided into approaches that use a mechanism to deal with concept drift [2,4,9,10] and approaches that do not explicitly use a mechanism to detect drifts [5,3]. The former approach uses some measure related to the accuracy to handle drifts. This approach rebuilds the system once a drift is detected / confirmed, so they cannot handle recurrent or predictable drifts anyway. They suffer from non-accurate drift detections but respond quickly to changing concepts. The latter approach assigns weights to each base learner according to its accuracy, allows deletion of poor performing classifiers and adds newly learnt classifiers. These approaches take longer time to recover from drifts but give more accurate results. The latter online approach would give better accuracy to handle changing concepts but the running time and memory requirements would be slightly more. So our work would use the concept of diversity on the latter online approach which should give better accuracy as compared to the approach where diversity was used on the former online approach as in DDD [8].

4 Literature Survey

In this section, we will give you the brief outline of the various online ensemble approaches that have been developed so far to handle concept drifts in data streams.

4.1 DWM: Dynamic Weighted Majority [5]

DWM is an online ensemble method for handling concept drift in the incoming data stream, by maintaining weighted pool of experts. It dynamically creates and removes

the experts in response to changes in its global performance (based on weighted majority voting), on the new training example. DWM outperforms other incremental learners that maintain and use previously encountered examples or an ensemble that employs an un-weighted or fixed-size ensemble of experts. The parameter p that controls the removal and creation of experts helps DWM in dealing with drifts in case of large and noisy data sets. DWM converges more quickly to the new target concepts than those that replace un-weighted learners.

4.2 DDM: Drift Detection Method [4]

The main idea behind DDM is to control the online error-rate of the algorithm. DDM adopts the dynamic window structure which is reduced when the error rate increases and increased when there is a reduction in error rate. It uses a warning level which states a possibility of a context change and a drift level which guarantees of a concept change. DDM is robust to false alarms. The method is very simple and is computationally efficient. However, this approach does not use any previous learning so they cannot be of any advantage in-case of recurrent drifts or predictable drifts. DDM does not perform well in-case of concepts which have a slow gradual change.

4.3 EDDM: Early Drift Detection Method [2]

To overcome this limitation of DDM, EDDM was proposed .It is based on the estimated distribution of the distances between classification errors as against error-rate in DDM. DDM and EDDM, both react quickly and reach low error rates on datasets with abrupt concept change. In case of noisy data, EDDM is more sensitive than DDM, detects changes very fast and improves the performance even when the base algorithm does not support noise. In case of slow gradual change, EDDM reacts before and more times than DDM.

4.4 AddExp :Addictive Expert Ensembles [6]

AddExp adds a new classifier whenever the system output is not correct, whose weight is the total weight of the ensemble times a constant $\gamma \in (0, 1)$. Two pruning methods were proposed: oldest first and weakest first. AddExp responds to sudden changes quickly. It does not consider the occurrence of recurring concepts and could not perform well for gradual changes because the new classifiers could not replace the old classifiers quickly.

4.5 ACE: Adaptive Classifier-Ensemble [9]

ACE is an adaptive classifier ensemble that uses an online classifier, a set of batch classifiers, and a drift detection mechanism to handle mainly recurrent drifts. When a new training example arrives, the online classifier is trained with this new example. However, the batch classifiers are not updated and they can be easily called for when re-curent drifts are encountered.

ACE, DWM and AddExp respond to sudden changes very quickly. However, this method does not have any classifier pruning mechanism, and any mechanism to sufficiently reduce the interference from old classifiers.

4.6 Enhanced ACE [10]

To overcome the limitations of ACE, this enhanced version of ACE was introduced. The basics of this algorithm are the same as ACE except it has an improved weighting method that is used to get the weighted majority vote of the outputs of all the classifiers. The method has also introduced a pruning strategy for classifiers, to improve the predictive accuracy for the new incoming examples. ACE responds well to sudden changes more quickly and more accurately than the original version and the added pruning method helped it to retain the useful classifiers.

4.7 STEP: Detection with Statistical Test of Equal Proportions [11]

STEPD detects concept drift by monitoring the predictive accuracy of a single online classifier. STEPD compares the two predictive accuracies: the overall accuracy from the beginning of the learning, and the accuracy of recent examples after concept drift, by using statistical test of equal proportions. STEPD performed the best for sudden changes. It detected concept drift very quickly and accurately on various data sets. Moreover, STEPD was comparable to EDDM for gradual changes

4.8 Todi: Two Online Classifiers for Learning and Detecting Concept Drift [1]

Todi was introduced to reduce the impact of false alarms. It uses two online classifiers for learning and detecting drifts. One of the classifiers is reinitialized and the other one is not, after the drift is detected. Todi uses only one significance level and does not store examples in short-term memory. It is robust to false alarms and gives quick and accurate drift detection. It performed the best for both sudden and gradual changes but does not consider the occurrence of recurring concepts.

4.9 DDD: Diversity for Dealing with Drifts [8]

The first time, when the concept of varying diversity levels between ensembles was used to handle concept drift, was in the work of DDD [8]. It used the drift detection method, that used a mechanism to explicitly handle drifts and that was EDDM. DDD was designed to be more robust to false alarms and has faster recovery from drifts and achieves improved accuracy in presence of drifts. In case of false alarms, it outperforms EDDM. DDD has always higher accuracy than DWM, both in presence as well as absence of drifts. However, there were many limitations with this approach. First, a measure related to the accuracies had to be decided, and it could not meet the constraints of lower memory requirements of a perfect online algorithm. Size of ensemble was fixed and had to be pre-decided and this approach could not be any way used for

dealing with re-current or predictable drifts. The approach also suffered from non-accurate drift detections and could not benefit more from the high diversity ensemble, before drift detection.

5 Problem Statement and the Proposed Approach

Develop a new online ensemble methodology to handle concept drift in data streams using the concept of Diversity. Our work would be the first approach where the concept of diversity would be applied on the online approach that does not explicitly use a mechanism to handle drifts. This approach would overcome the limitations of the earlier approaches. The main idea would be that before the detection of drift a low diversity and a high diversity ensemble were maintained. Both the ensembles were used for training but only low diversity ensemble was used for system predictions. After the drift was detected, new high and new low diversity ensembles would be created and the earlier low and high diversity ensembles were denominated as old low and old high diversity ensembles. The old high diversity ensemble then started learning with low diversity to adapt to the new concept.

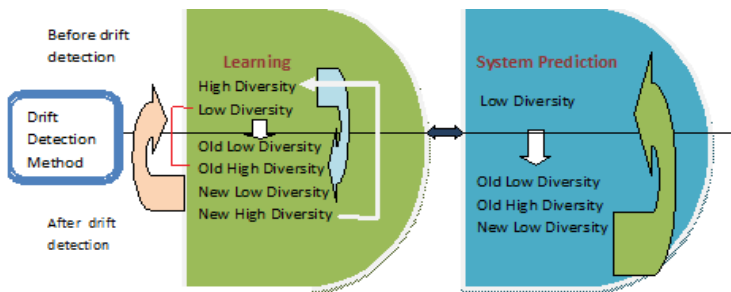


Fig. 1. Framework of Our Proposed Approach

Both the old and the new ensembles performed learning but the system predictions were the weighted majority vote of the output of old low, old high and new low diversity ensembles. We will be using an extension of DWM, EDWM as a drift detection method. The framework of our approach is as shown in Fig. 1.

6 Results Achieved

We have done the complete literature survey (Phase 1) of the various online methodologies to handle concept drift in data streams. Further, the drift detection method to be used (Phase 2); EDWM has been framed and implemented using various datasets. EDWM, showed similar accuracy as DWM in lesser time when implemented on Hyperplane Dataset and the results on SEA Concepts concluded that EDWM gave similar accuracy than DWM in same time constraint. EDWM showed better accuracy

than Weighted Majority algorithm [7], which was an incremental approach. It did not suffer from non-accurate drift detections and dynamically updated the size of ensembles which was fixed in case of EDDM. The results have been communicated as a paper in one of the International conferences.

7 Future Work

Phase 3: Create Ensembles using Online Bagging using Poisson distribution.

Phase 4: Design the complete algorithm using EDWM as drift detection method, Modified Online Bagging for ensemble creation and Naïve Bayes Classifiers for learning.

Phase 5: Implementation in Matlab or Massive Online Analysis (MOA).

Phase 6: Testing on various datasets with different types of drifts.

Phase 7: Comparison with earlier online ensemble approaches.

Phase 8: Extending the approach using Online Boosting for ensemble creation or other classifiers for ensemble learning.

Our research has been completed till Phase 2 and we would try to meet the timelines for the future work to submit my dissertation in the expected time.

References

1. Hokkaido University, <http://lis2.huie.hokudai.ac.jp/~knishida/paper/nishida2008-dissertation.pdf>
2. Baena-García, M., Avila, J., Del Campo, F.R., Bifet, A.: Early Drift Detection Method. In: Proc. of the 4th ECML PKDD International Workshop on Knowledge Discovery from Data Streams, Berlin, Germany, pp. 77–86 (2006)
3. Stanley, K.O.: Learning concept drift with a committee of decision trees. Technical Report, Department of Computer Sciences, University of Texas at Austin, Austin, USA (2003)
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. Kolter, J.Z., Maloof, M.A.: Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts. *The Journal of Machine Learning Research* 8, 2755–2790 (2007)
6. Kolter, J.Z., Maloof, M.A.: Using additive expert ensembles to cope with concept drift. In: Proceedings of 22rd International Conference on Machine Learning, Bonn, Germany, pp. 449–456 (2005)
7. Littlestone, N., Warmth, M.K.: The Weighted Majority algorithm. *Information and Computation* 108, 212–261 (1994)
8. Minku, L.L., Yao, X.: DDD: A New Ensemble Approach for Dealing with Concept Drift. *IEEE Transactions on Knowledge and Data Engineering* 24(4), 619 (2012)
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. Nishida, K., Yamauchi, K.: Adaptive classifiers-ensemble system for tracking concept drift. In: Proc. 6th International Conference on Machine Learning and Cybernetics, Hong Kong, pp. 3607–3612 (2007)
11. Nishida, K., Yamauchi, K.: Detecting concept drift using statistical testing. In: Corruble, V., Takeda, M., Suzuki, E. (eds.) DS 2007. LNCS (LNAI), vol. 4755, pp. 264–269. Springer, Heidelberg (2007)

Author Index

- Adelfio, Marco D. 245
Ait Ameer, Yamine 1
Akli Astouati, Karima 17
Algergawy, Alsayed 267
Amous, Ikram 121
Andrzejewski, Witold 1
Angryk, Rafal A. 27, 87, 151
Apiletti, Daniele 169
Appel, Ana Paula 177
Aramburu, María José 319
Augustyn, Dariusz Rafal 215
- Baba-Hamed, Latifa 297
Banda, Juan M. 27, 87, 151
Basso, Simone 203
Baumgärtel, Philipp 79
Bellatreche, Ladjel 1, 141, 225
Ben Djemaa, Raoudha 121
Benkrid, Soumia 141
Benslimane, Djamal 129
Berlanga, Rafael 319
Bessagnet, Marie-Noëlle 257
Bhatia, M.P.S. 389
Boniewicz, Aleksandra 105
Bourai, Safia Bal 277
Boustia, Narhimene 17
Breß, Sebastian 225
Buscaldi, Davide 257
- Camillo, Furio 329
Catania, Barbara 1
Cerquitelli, Tania 1
Chaturvedi, Amrita 307
Cherif, Sihem 121
- Chiusano, Silvia 1
Cuzzocrea, Alfredo 141
- Damigos, Matthew 69
de Carvalho, Veronica Oliveira 45
de Gemmis, Marco 383
De Martin, Juan Carlos 203
de Padua, Renan 45
de Souza Serapião, Adriane Beatriz 45
Dessi, Nicoletta 351
- Eldosouky, Ali 267
- Farina, Jacopo 159
Favre, Cécile 361
Fellegara, Riccardo 373
Forno, Fabio 169
- Ganesan Pillai, Karthik 27, 151
Garau, Gianfranco 351
García-Moya, Lisette 319
Gargouri, Faïez 287
Garza, Paolo 187
Gergatsoulis, Manolis 69
Ghedira Guegan, Chirine 129
Golfarelli, Matteo 1
Grandi, Roberto 339
Grimaudo, Luigi 187
Guebaili Djider, Ratiba 17
Guerrini, Giovanna 1
Gurský, Peter 37
- Heimel, Max 225
Holubová (Mlýnková), Irena 113
Hruschka, Estevam Rafael Junior 177

- Jakawat, Wararat 361
 Kaczmarek, Krzysztof 1, 53, 235
 Kalinichenko, Leonid 61
 Kämpf, Mirko 1
 Kemper, Alfons 1
 Khellaf, Faiza 277
 Kirikova, Marite 97
 Kovalev, Dmitry 61

 Lauer, Tobias 1
 Lenz, Richard 79
 Liberati, Caterina 329
 Llidó, Dolores M. 319
 Loudcher, Sabine 361

 Maalej, Maha 287
 Margara, Paolo 187
 Masala, Enrico 203
 Mašiček, Viktor 113
 Mazuran, Mirjana 159
 McInerney, Patrick 87, 151
 Moawed, Seham 267
 Mokadem, Riad 129
 Mokhtari, Aicha 17, 277
 Morvan, Franck 129
 Mtibaa, Achraf 287

 Namber, Magloire 297
 Navarro, Lucas Fonseca 177
 Nepote, Nicolò 187
 Neri, Federico 339
 Nouioua, Farid 17
 Novikov, Boris 1

 Palpanas, Themis 1
 Piccolo, Elio 187

 Plitsos, Stathis 69
 Pokorný, Jaroslav 1
 Przymus, Piotr 53, 235
 Pudane, Mara 97

 Quintarelli, Elisa 159, 193

 Rabosio, Emanuele 193
 Ricci, Giuseppe 383
 Rizzi, Stefano 1
 Royer, Albert 257

 Saake, Gunter 225, 267
 Sallaberry, Christian 257
 Samet, Hanan 245
 Sankaranarayanan, Jagan 245
 Sarhan, Amany 267
 Schuh, Michael A. 27, 87, 151
 Semeraro, Giovanni 383
 Servetti, Antonio 203
 Sidhu, Parneeta 389
 Siegmund, Norbert 225
 Stencel, Krzysztof 105
 Stupnikov, Sergey 61
 Šumák, Martin 37

 Teitler, Benjamin E. 245
 Tenschert, Johannes 79
 T.V., Prabhakar 307

 Vakali, Athena 1
 Vovchenko, Alexey 61

 Warchal, Lukasz 215
 Wiśniewski, Piotr 105
 Wylie, Tim 27, 87