SEBD 2012

Proceedings of the 20th Italian Symposium on Advanced Database Systems

June 24—27, 2012

Venice, Italy

Nicola Ferro
Letizia Tanca

Nicola Ferro Letizia Tanca (Eds.)

Proceedings of 20th Italian Symposium on Advanced Database Systems

SEBD 2012

June 24–27, 2012
Venice, Italy

Volume Editors

Nicola Ferro
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Padova
Via G. Gradenigo, 6/B
35131 Padova, Italy
ferro@dei.unipd.it

Letizia Tanca
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci, 32
20133 Milano, Italy
tanca@elet.polimi.it

ISBN 978-88-96477-23-6

Preface

This volume collects the papers and posters selected for presentation at the Twentieth Italian Symposium on Advanced Database Systems (SEBD 2012), held in Venice from the 24^{th} to the 27^{th} of June 2012.

The first edition of SEBD was held in 1993 as an outcome of a nationwide project funded by the Italian National Research Council, called Logidata+. The success of the event encouraged the organizers to repeat the experience year after year, until the symposium has became the annual gathering where the Italian database research community discusses not only original research, but also trends, policies, funding, assessment, and other cogent problems of research and teaching in computer science in general.

This twentieth birthday of SEBD represents a major achievement of the Italian database research community, which has shown its ability to react to, and cope with, the many research challenges that have been arising during these twenty years. Together, we have witnessed the activities and liveliness of a research community that has involved "generations" of researchers, offering to young researchers, through SEBD, a stage for discussing and confronting ideas.

Traditionally, the database research community has mainly concentrated on the methodologies, techniques, and technologies for data management. While in the past the role of data was recognized as central mostly in the organization of business activities, in the last decade this equilibrium has been completely overturned, and data-related research and development problems are involved in every individual and collective activity of society.

The main two phenomena that have nurtured this trend are the availability of information through the Web and the possibility to access resources anywhere and at any time through mobile, portable devices. Now everybody can both consume and produce information by means of various sorts of easy interaction means: social computing involves people and stimulates them to the use of sophisticated applications and devices; the pervasiviness of computation means (very small, cheap, and low power devices, connected through wireless networks) relieves people of a cumbersome interaction with passive tools, by embedding the processing power in the environment, often anticipating the user's needs; new forms of information, like scientific or multimedia data, require equally new forms of storage and processing. These are only examples of the most various information, from multimedia data streams and storage systems to semantic-web knowledge linked all over the Web, to natural language information that has to be automatically understood.

The availability of so huge amounts of very different kinds of data ("big data") is changing our attitude towards them, requiring more and more sophisticated analysis means. Therefore, a data-centric vision of the world is actually key for advancing in such a demanding scenario. The vision of an inclusive society for the future requires effective, usable, and secure methodologies and techniques

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

for data production, management and analysis, which will enable us to give semantics to, and make sense of, the enormous quantity of information which is continuously produced and very often ignored or ill-used.

In this scenario, two apparently contrasting phenomena take place: on the one hand, the research becomes more and more specialized as the results of investigation on long-term research problems generate the need to delve deeper into more specific issues; on the other hand, different disciplines join forces with ours, displaying their full power to answer demands posed by brand-new applications and visionary technologies.

These two trends are shown by the topics presented in this edition of SEBD: Knowledge Discovery and Extraction; Data Streams and Uncertainty; Semantic Web and Interoperability; Data Mining; Information Access and Retrieval; Query Answering; Logic and Deductive Databases; Classification and Clustering; and, Integration and Information Exchange.

37 contributions were originally submitted, of which 9 full papers and 28 extended abstracts. Most of the contributions came from universities and the CNR (Italian National Research Council). Each submission was evaluated by three independent referees and the selection process led to the acceptance of 4 full papers, 23 extended abstracts, and 7 posters. Posters, which were introduced for the first time in the 2011 edition of SEBD, represent a better mean for either discussing new emerging ideas or presenting more consolidated research in a more interactive and less time-constrained fashion.

Besides paper and poster presentations, the programme also featured two keynote talks which highlighted future research directions. The first one by prof. Timos Sellis (Research Center "Athena" and National Technical University of Athens, Greece) explored "Personalization in Web Search and Data Management"; the second one by Dr. Krishna P. Gummadi (Max Planck Institute for Software Systems (MPI-SWS), Germany) dealt with "Extracting Relevant and Trustworthy Information from Microblogs". Finally, one tutorial by prof. Maarten de Rijke (ISLA, University of Amsterdam, The Netherlands) concerned "Log File Analysis and Mining". Altogether, the invited talks and the tutorial not only offered an interesting perspective on compelling research challenges but also provided bridges connecting typical database themes with topics of nearby disciplines, such as information access and retrieval.

Moreover, consistently with the already-mentioned custom of discussing important issues which are transversal with respect to the technical topics, the symposium hosted two "Community Think-Tank" sessions, during which the fundamental topic of research evaluation was discussed.

We wish to thank all the authors who submitted papers and all the conference participants. We are also grateful to the members of the Programme Committee and the external referees for their thorough work in reviewing submitted contributions with expertise and patience, and to the members of the SEBD Steering Committee for their support in the various decisions that had to be taken during the event organization. Finally, we wish to thank the members of the Local

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Organization Committee for their willingness and contribution in dealing with the many aspects which are involved by the logistics of a conference.

We gratefully acknowledge the patronage of several institutions: Regione del Veneto (http://www.regione.veneto.it/); Università degli Studi di Padova (http://www.unipd.it/); and, Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Padova (http://www.dei.unipd.it/).

We also express our gratitude to the many sponsoring organizations and initiatives for their significant and timely support: Associazione Italiana per l'Informatica ed il Calcolo Automatico (AICA, http://www.aicanet.it/); Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Padova; PROMISE (FP7 NoE, contract n. 258191, http://www.promise-noe.eu/); and, CULTURA (FP7 STREP, contract n. 269973, http://www.cultura-strep.eu/).

This SEBD edition comes only a few weeks after the earthquake that hit the Italian region of Emilia Romagna in May 2012. The SEBD community wishes to express its empathy with the many region inhabitants who have lost their homes, jobs and even lives, making good omens for a quick reconstruction of the social, productive, and cultural life in that area.

June 2012

Nicola Ferro
Letizia Tanca

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Organization

SEBD 2012 is organized by the Information Management Systems (IMS) research group of the Department of Information Engineering (DEI) of the University of Padua, Italy.

General Chair

Letizia Tanca, Politecnico di Milano

Program Chair

Nicola Ferro, Università degli Studi di Padova

Local Organization Chair

Gianmaria Silvello, Università degli Studi di Padova

Local Organization Committee

Debora Leoncini, Università degli Studi di Padova
Ivano Masiero, Università degli Studi di Padova
Simone Peruzzo, Università degli Studi di Padova

Program Committee

Giuseppe Amato, Istituto di Scienza e Tecnologie dell'Informazione, CNR Pisa
Elena Baralis, Politecnico di Torino
Carlo Batini, Università degli Studi di Milano-Bicocca
Devis Bianchini, Università degli Studi di Brescia
Andrea Calì, University of London, Birkbeck College
Cinzia Cappiello, Politecnico di Milano
Michelangelo Ceci, Università degli Studi di Bari
Augusto Celentano, Università Ca' Foscari Venezia
Paolo Ciaccia, Università di Bologna
Valter Crescenzi, Università degli Studi Roma Tre
Paolino Di Felice, Università degli Studi dell'Aquila

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Claudia Diamantini, Università Politecnica delle Marche
Alfio Ferrara, Università degli Studi di Milano
Enrico Franconi, Libera Università di Bolzano
Giorgio Ghelli, Università di Pisa
Giovanna Guerrini, Università degli Studi di Genova
Nicola Orio, Università degli Studi di Padova
Luigi Palopoli, Università della Calabria
Antonella Poggi, Università degli Studi di Roma "La Sapienza"
Carlo Sartiani, Università degli Studi della Basilicata
Domenico Ursino, Università Mediterranea di Reggio Calabria
Yannis Velegrakis, Università degli Studi di Trento
Maurizio Vincini, Università degli Studi di Modena e Reggio Emilia

External Reviewers

Domenico Beneventano, Università degli Studi di Modena e Reggio Emilia
Mirko Bronzi, Università degli Studi Roma Tre
Luca Cagliero, Politecnico di Torino
Dario Colazzo, Université Paris Sud
Fabio Fassetti, Università della Calabria
Alessandro Fiori, Politecnico di Torino
Sergio Flesca, Università della Calabria
Giancarlo Fortino, Università della Calabria
Paolo Garza, Politecnico di Milano
Lorenzo Genta, Università degli Studi di Milano
Alberto Grand, Politecnico di Torino
Gianluca Lax, Università Mediterranea di Reggio Calabria
Lucrezia Macchia, Università degli Studi di Bari
Giuseppe Manco, Istituto di Calcolo e Reti ad Alte Prestazioni, CNR Cosenza
Andrea Marin, Università Ca' Foscari Venezia
Andrea Maurino, Università degli Studi di Milano-Bicocca
Marek Maurizio, Università Ca' Foscari Venezia
Michele Melchiori, Università degli Studi di Brescia
Paolo Merialdo, Università degli Studi Roma Tre
Marco Mesiti, Università degli Studi di Milano
Luigi Moccia, Istituto di Calcolo e Reti ad Alte Prestazioni, CNR Cosenza
Antonino Nocera, Università Mediterranea di Reggio Calabria
Marius-Octavian Olaru, Università degli Studi di Modena e Reggio Emilia
Matteo Palmonari, Università degli Studi di Milano-Bicocca

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Gianvito Pio, Università degli Studi di Bari
Giovanni Quattrone, University College London
Domenico Rosaci, Università Mediterranea di Reggio Calabria
Donatello Santoro, Università degli Studi della Basilicata
Daniela Stojanova, Jozef Stefan Institute Ljubljana

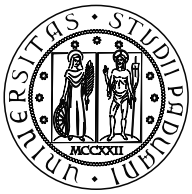
SEBD Steering Committee

Maristella Agosti, Università degli Studi di Padova
Antonio Albano, Università di Pisa
Paolo Atzeni, Università degli Studi Roma Tre
Sonia Bergamaschi, Università degli Studi di Modena e Reggio Emilia
Valeria De Antonellis, Università degli Studi di Brescia
Maurizio Lenzerini, Università degli Studi di Roma "La Sapienza"
Domenico Saccà, Università della Calabria
Fabio Schreiber, Politecnico di Milano
Letizia Tanca, Politecnico di Milano
Paolo Tiberio, Università degli Studi di Modena e Reggio Emilia

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Patronage

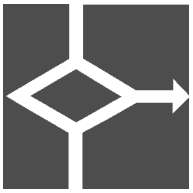
PATROCINIO
REGIONE DEL VENETO



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Sponsors



AICA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

PROMISE
www.promise-noe.eu

cultura
www.cultura-strep.eu

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Table of Contents

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Semantic Web and Interoperability

Integration and Information Exchange

Information Access and Retrieval

Query Answering

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Logic and Deductive Databases

Classification and Clustering

Data Mining

Posters

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Extracting Relevant and Trustworthy Information from Microblogs

Krishna P. Gummadi

Max Planck Institute for Software Systems (MPI-SWS), Germany
gummadi@mpi-sws.org

Microblogging sites like Twitter have emerged as a popular platform for exchanging real-time information on the Web. Twitter is used by hundreds of millions of users ranging from popular news organizations and celebrities to domain experts in fields like computer science and astrophysics and spammers. As a result, the quality of information posted in Twitter is highly variable and finding the users that are authoritative sources of relevant and trust-worthy information on specific topics (i.e., topical experts) is a key challenge. I will attempt to address this challenge in this two-part talk.

In the first part of the talk, I will focus on understanding and combating link farming activity in Twitter. Users, especially spammers, resort to link farming to acquire large numbers of follower links in the social network. Acquiring followers not only increases the size of a user's direct audience, but also contributes to the perceived influence of the user, which in turn impacts the ranking of the user's tweets by search engines. I will first discuss results from our recent studies investigating link farming activity in the Twitter network and then propose mechanisms to discourage the activity.

In the second part of the talk, I will focus on the problem of finding topic experts in Twitter. I will propose a new methodology that relies on the wisdom of the Twitter crowds. Specifically, we leverage Twitter Lists, which are often carefully created by individual users to include experts on topics that interest them and whose meta-data (List names and descriptions) provide valuable semantic cues to experts' domain of expertise. I will first describe how we mined List information to build Cognos, an expert search system for Twitter and then present results from a real-world deployment.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Personalization in Web Search and Data Management

Timos Sellis
(joint work with T. Dalamagas, G. Giannopoulos, and A. Arvanitis)

Research Center "Athena" and National Technical University of Athens, Greece
timos@imis.athena-innovation.gr

We address issues on web search personalization by exploiting users' search histories to train and combine multiple ranking models for result reranking. These methods aim at grouping users' clickthrough data (queries, results lists, clicked results), based either on content or on specific features that characterize the matching between queries and results and that capture implicit user search behaviors. After obtaining clusters of similar clickthrough data, we train multiple ranking functions (using Ranking SVM model), one for each cluster. Finally, when a new query is posed, we combine ranking functions that correspond to clusters similar to the query, in order to rerank/personalize its results.

We also present how to support personalization in data management systems by providing users with mechanisms for specifying their preferences. In the past, a number of methods have been proposed for ranking tuples according to user-specified preferences. These methods include for example top-k, skyline, top-k dominating queries etc. However, neither of these methods has attempted to push preference evaluation inside the core of a database management system (DBMS). Instead, all ranking algorithms or special indexes are offered on top of a DBMS, hence they are not able to exploit any optimization provided by the query optimizer. In this talk we present a framework for supporting user preference as a fist-class construct inside a DBMS, by extending relational algebra with preference operators and by appropriately modifying query plans based on these preferences.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Log File Analysis and Mining

Maarten de Rijke

ISLA, University of Amsterdam, Science Park 904
1098 XH Amsterdam, The Netherlands
derijke@uva.nl

Information retrieval is no longer just about matching the content of queries to the content of documents. For nearly two decades, links and link structure have been brought to bear on the information retrieval problem in a web setting. During the past five years, as part of the development of information retrieval algorithms, content and link analysis are increasingly being complemented with insights gleaned from observation of people and of people's interactions with information and the search engine.

One of the ways in which user search behaviors may be analyzed is through a transaction log analysis, which over the years has proved an apt method for the characterization of user behavior. Its strengths include its non-intrusive nature — the logs are collected without questioning or otherwise interacting with the user — and the large amounts of data that can be used to generalize over the cumulative actions taken by large numbers of users. It is important to note that transaction log analysis faces limitations: not all aspects of the search can be monitored by this method, for example, the underlying information need. It can also be difficult to compare across transaction log studies of different systems due to system dependencies and varying implementations of analytical methods. Comparability can be improved to some extent by providing clear descriptions of the system under investigation and the variables used.

Information retrieval has a long history of transaction log analysis, from early studies of the logs created by users of library online public access catalog systems to later studies of the logs of Web search engines. This was followed by the analysis of more specialized search engines and their transaction logs. For instance, authors have studied the behavior of users of a blog search engine through a log file analysis and examined the difference between the vocabularies of queries, social bookmarking tags, and online documents. Three frequently used units of analysis have emerged from the body of work: the *session*, the *query*, and the *term*, though the definition of each unit may vary across studies.

In the tutorial, I will provide a number of examples of log file studies as well as the type of knowledge that can be obtained by studying log files: about people's information behavior, about experimental evaluation of search engines, and about online optimizations of search engines.

The tutorial is based on joint work with Richard Berendsen, Katja Hofmann, Bouke Huurnink, Bogomil Kovachev, Edgar Meij, Gilad Mishne, Evangelia-Paraskevi Nastou, Wouter Weerkamp, and Shimon Whiteson.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Sulla Classificazione delle Sedi di Pubblicazione nella Valutazione della Produzione Scientifica

Giansalvatore Mecca, Marcello Buoncristiano, and Donatello Santoro

Dipartimento di Matematica e Informatica
Università della Basilicata – Potenza – Italy
http://db.unibas.it/valutazione

1 Introduzione e Motivazioni

Questo lavoro discute alcune tecniche per la valutazione della produzione scientifica. Si tratta della versione ridotta di un lavoro più ampio (Mecca, 2012); rimandiamo alla versione estesa per tutti i dettagli di carattere tecnico che in questa sede sono stati omessi.

Il problema della valutazione della produzione scientifica è di grande attualità nelle Università italiane. E' di questi giorni l'avvio delle procedure di valutazione nazionale della qualità della ricerca relativa al quadriennio 2004-2010, e la pubblicazione del D.M. n. 19 del 27 gennaio 2012 (Ministero dell'Università e della Ricerca, 2012) che introduce il sistema nazionale di valutazione del sistema universitario e obbliga le università a dotarsi di un proprio sistema di autovalutazione basato su "parametri oggettivi, volti a misurare in ogni momento l'efficienza e l'efficacia della didattica e della ricerca messa in atto dai singoli atenei e a stimolare la competitività e la qualità degli stessi".

E' quindi ragionevole attendersi che nell'arco dei prossimi mesi tutte le università italiane dovranno dotarsi di modelli e procedure per la misurazione della "performance" dei docenti (professori e ricercatori). In questo articolo, ci concentriamo su uno degli aspetti che le Università sono chiamate a valutare, ovvero la ricerca, e più specificamente, sulla valutazione della produzione scientifica.

Le attività di valutazione di cui trattiamo hanno caratteristiche ben precise: (*a*) considerano la produzione scientifica dei docenti dell'ateneo in un periodo di riferimento, tipicamente tre o cinque anni; (*b*) vengono condotte periodicamente, per esempio una volta all'anno, e su larga scala, perché riguardano tutti i docenti dell'Ateneo; (*c*) richiedono di confrontare la produzione di docenti di aree anche molto diverse tra di loro – dalla matematica alla sociologia – per esempio a scopo di ripartizione di risorse; (*d*) devono essere concluse in tempi ragionevolmente brevi per servire allo scopo per cui sono adottate; (*e*) possono contare su budget limitati o addirittura assenti.

E' importante rimarcare le differenze tra valutazioni dei singoli su scala di Ateneo ed iniziative con scopi e finalità diverse. L'ANVUR, ad esempio, sta conducendo un'attività di valutazione che non riguarda i singoli, ma le strutture. Si tratta di un processo di lunga durata (durerà più di un anno), con un notevole budget economico. Non è possibile, di conseguenza, pensare di adottare esattamente le stesse tecniche per i nostri scopi.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

In effetti, le caratteristiche del processo di cui ci occupiamo costringono a scelte sostanzialmente obbligate relativamente al metodo da adottare per condurre la valutazione. E' noto il fatto che esistono tre tecniche principali per la valutazione della produzione scientifica. (1) il "peer reviewing", ovvero la valutazione da parte di revisori anonimi; (2) l'uso di indicatori bibliometrici basati sul numero di citazioni ricevute dai lavori; (3) l'uso della qualità delle sedi di pubblicazione.

Nell'ambito delle procedure di valutazione di cui parliamo, il "peer reviewing"- tecnica, peraltro, non esente da limiti, come discusso ampiamente nella letteratura recente (Casati, 2009) – è chiaramente inapplicabile, per via dei vincoli di tempo e di risorse.

E' delicato anche l'utilizzo esclusivo di indici bibliometrici basati sul numero di citazioni, per varie ragioni. Per cominciare, questi indici sono più solidi se utilizzati per analizzare la produzione scientifica in un periodo medio-lungo; mostrano una certa fragilità se applicati a periodi più brevi e recenti. Inoltre, è dimostrato (Iglesias, 2007) che le diverse comunità scientifiche hanno stili e pratiche di citazione completamente diversi gli uni dagli altri, e di conseguenza che è estremamente delicato confrontare comunità lontane sulla base di questi indici.

Alla luce di queste considerazioni, nel seguito di questo lavoro ci concentreremo esclusivamente sul terzo metodo, ovvero la valutazione dei prodotti di ricerca attraverso la valutazione della qualità delle corrispondenti sedi di pubblicazione. E' possibile pensare che questa tecnica sia affiancata in modo più o meno significativo dall'utilizzo di indicatori basati sulle citazioni dei lavori, dei quali però non discuteremo ulteriormente.

In sintesi, nell'approccio di cui trattiamo, il valore di un lavoro scientifico – per esempio un articolo pubblicato su rivista o negli atti di un convegno – viene valutato sulla base della "*classe di merito*" della corrispondente rivista o del corrispondente convegno. La classe di merito assume normalmente un valore pari a A, B, C, D, per indicare le sedi di primo piano, quelle di secondo piano eccetera.

E' facile immaginare come, in questo approccio, l'aspetto centrale della valutazione sia l'adozione di classificazioni affidabili delle sedi di pubblicazione.

Il problema centrale affrontato in questo lavoro è quello della combinazione tra diversi ranking della stessa sede di pubblicazione. Vengono discusse tre diverse classificazioni, una per le riviste, una per i convegni, una per le case editrici. In tutti i casi, le classificazioni sono state generate combinando ranking esistenti di carattere internazionale e nazionale. A questo scopo, sono stati predisposti diversi strumenti informatici, a supporto di un processo complesso di estrazione, pulitura e integrazione dei dati.

ABDOMINAL IMAGING		Aggregate ranking B/C ?		
ISI	GASTROENTEROLOGY & HEPATOLOGY	ABDOMINAL IMAGING	0942-8925	C
ISI	RADIOLOGY, NUCLEAR MED. & MED. IMAGING	ABDOMINAL IMAGING	0942-8925	C
Scopus	Medicine	Abdominal Imaging	1432-0509	B
ERA		Abdominal Imaging	0942-8925	C

Figura 1 Gruppo di Valori di Ranking per la Rivista "Abdominal Imaging". Due provengono da categorie ISI, uno dall'Area Scopus "Medicine", l'ultimo dalla classificazione ERA

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Al termine di questo passo, per ciascuna sede di pubblicazione è stato costruito un gruppo di valori di ranking, come mostrato in Figura 1 per la rivista "Abdominal Imaging". Il passo successivo è quello cruciale, e consiste nell'assegnare a ciascuna sede di pubblicazione un valore di ranking, omogeneizzando opportunamente quelli presenti nelle classificazioni di partenza.

2 Funzioni di Voto e di Consenso

Il problema che intendiamo affrontare è quello della definizione di strategie per l'integrazione di classificazioni alternative della stessa sede di pubblicazione, ad esempio una rivista. Dall'unione delle diverse classificazioni, ciascuna sede di pubblicazione riceve un insieme di *valori di ranking*, come mostrato in Figura 1. I valori di ranking provengono da un dominio discreto, enumerabile ed ordinato. In prima istanza, supponiamo che i valori possibili siano, in ordine decrescente, {A+, A, B, C, D, NC}, dove NC è il "valore nullo" del dominio che indica una sede di pubblicazione non classificabile.

Si noti che l'insieme dei valori di ranking all'interno di una classificazione rappresenta una variabile aleatoria a valori discreti. Le diverse variabili aleatorie possono difficilmente essere considerate indipendenti, dal momento che tutte hanno l'obiettivo di misurare il valore scientifico delle sedi di pubblicazione, tipicamente attraverso una stima della relativa reputazione e del relativo impatto.

Una *funzione di ranking* associa ad un gruppo di valori di ranking un valore di ranking aggregato, e rappresenta una strategia per combinare i diversi ranking disponibili per una stessa sede di pubblicazione.

Nella letteratura sono documentate varie soluzioni al problema. Alcune di queste sono basate puramente sull'attribuzione di una priorità alle sorgenti di classificazione: in caso di valori multipli di ranking prevale quello della sorgente a maggiore priorità. Un esempio di questa strategia è applicato dall'Università di Bologna, che ha sviluppato una propria classificazione, ma in tutti i casi in cui una rivista è indicizzata dall'ISI dà priorità al ranking basato sull'impact factor.

Altre soluzioni adottano approcci più sofisticati. In particolare, è in linea di principio possibile adottare funzioni di carattere numerico che combinino i diversi valori del gruppo. Questo approccio, però, nasconde numerose insidie. Per cominciare, come discusso precedentemente, si tratta di combinare i valori di variabili aleatorie non indipendenti. E' inoltre noto che funzioni di aggregazione semplici, come la media dei valori numerici corrispondenti ai valori di ranking, possono condurre ad errori grossolani (Thompson, 1993).

Queste considerazioni, che mettono in luce le insidie dell'operazione che stiamo conducendo, suggeriscono l'adozione di funzioni di ranking alternative. In effetti, è relativamente semplice immaginare funzioni di ranking che facciano al nostro scopo. Due esempi sono la funzione "*max*", che associa alla sede di pubblicazione il valore più alto tra quelli del gruppo, e la funzione "*min*", che restituisce il valore più basso diverso da NC.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nel contesto della valutazione della ricerca, l'utilizzo esclusivo di queste due funzioni sembra inopportuno; l'esperienza mostra infatti che non è infrequente l'esistenza di gruppi di ranking ad alta distanza. La *distanza* di un gruppo di valori di ranking è il valore assoluto della differenza tra il valore più piccolo e quello più grande appartenenti al gruppo, escludendo il valore NC[1]. Ad esempio, nel gruppo [A+, A, B, B, C] la distanza è 3. In un gruppo del genere, sembra del tutto arbitrario l'utilizzo tanto della funzione "max", che selezionerebbe il valore di ranking aggregato A+, che della funzione "min", che selezionerebbe il valore di ranking aggregato C.

Di conseguenza, in questo lavoro adotteremo un approccio diverso, basato sull'utilizzo di *funzioni di voto e di consenso*. Il modello sottostante al nostro approccio è quello dell'elezione. Ciascuna categoria esprime per una sede di pubblicazione un *voto* relativamente al suo ranking. Le funzioni di nostro interesse cercano i valori che, sulla base del ranking, raggiungono il massimo supporto.

La prima, naturale funzione che introduciamo è la funzione "*majority*". La funzione seleziona in un gruppo il valore di ranking che riceve un numero di voti pari alla *maggioranza assoluta*. Ad esempio, nel gruppo [A+, B, B, B, C], la funzione seleziona il valore B, che ha ricevuto 3 voti sui cinque espressi. Nel gruppo [A, A, B, NC], la funzione seleziona il valore A, che ha ricevuto 2 voti sui 3 espressi (in questo caso escludiamo dal computo della maggioranza il valore NC, che è considerato un "non voto"). Si noti che la funzione "majority" non è definita per tutti i gruppi. Ad esempio, per il gruppo [A, A, B, C], non restituisce alcun valore perché non c'è maggioranza assoluta. Lo stesso vale per il gruppo [A, A, B, B]. In questo caso assumeremo che il valore restituito dalla funzione sia NC.

Come è comune nell'utilizzo di funzioni di voto (Wikipedia - MF) per dirimere la maggioranza nel caso di voti in numero pari, è possibile introdurre un *pregiudizio* a favore o a sfavore della maggioranza. La ragione di questo fatto è che ottenere maggioranze assolute su gruppi di dimensione pari e di piccole dimensioni è più difficile: in un gruppo di 6 valori, la maggioranza assoluta richiede 4 voti, il che, in percentuale, è molto più dei 4 voti necessari in un gruppo di 7 valori.

Definiamo di conseguenza la variante "*mildMajority*" della funzione "majority", introducendo un pregiudizio a favore della maggioranza; più precisamente, diciamo che un valore ha la maggioranza assoluta se si verificano tutte le seguenti condizioni: (a) ha ricevuto la maggioranza relativa dei voti[2]; (b) il suo numero di voti aumentato di

[1] Per differenza tra due valori in un insieme discreto ordinato intendiamo la differenza tra la posizione nell'ordinamento del primo valore e quella del secondo valore.

[2] Un valore riceve la maggioranza relativa dei voti quando è il più votato in assoluto (non c'è un altro valore che ha riportato lo stesso numero di voti). In termini statistici, questo implica che il valore è la *moda* (Wikipedia - Moda) della distribuzione, e che la distribuzione è di tipo *unimodale*; aver ricevuto la maggioranza relativa non implica il raggiungimento della maggioranza assoluta; ad esempio, nel gruppo [A, A, B, C, D], il valore A ha ricevuto la maggioranza relativa dei voti, ma non la maggioranza assoluta, che richiederebbe 3 voti. Nel gruppo [A, A, B, B], come nel gruppo [A, B] nessun valore riceve la maggioranza relativa, e di conseguenza la maggioranza assoluta. La condizione imposta sulla funzione è particolarmente importante per questo tipo di distribuzioni, dette *bimodali*, per evitare che la funzione scelga arbitrariamente uno dei due valori più frequenti.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

0,1 supera la metà del numero di voti espressi nel gruppo. Si noti che il pregiudizio è tale per cui, nel caso di numero di voti dispari non c'è nessuna differenza tra "majority" e "mildMajority"; viceversa, la differenza è significativa in caso di numero di voti pari. Nel caso discusso sopra, [A, A, B, C], "majority" vale NC, mentre "mildMajority" vale A. Resta, viceversa, NC il valore nel caso dei gruppi [A, A, B, B], [A, B, C] e [A, B].
Per risolvere opportunamente anche questi gruppi, è possibile ulteriormente ammorbidire la funzione, passando dal concetto di voto al concetto di *consenso*. Abbiamo finora assunto che una categoria di classificazione "vota" per un valore di ranking in tutti i casi in cui esprime esattamente quel valore. Diciamo ora che, votando per un valore, la categoria esprime anche implicitamente *consenso* per tutti i valori inferiori. In altri termini, votando per A, si esprime consenso per A, ma anche per B, C e D.
La funzione "*majorityConsensus*" (e la sua variante "*mildMajorityConsensus*") restituisce il valore più alto di un gruppo per cui c'è un consenso superiore alla metà dei voti (ovvero, il consenso aumentato di 0,1 supera la metà dei voti).
Le funzioni "majorityConsensus" e "mildMajorityConsensus" consentono di risolvere alcuni dei gruppi per cui non c'è maggioranza dei voti, ad esempio [A, B, C], con il valore B.
Ulteriori dettagli relativi alle funzioni utilizzate sono descritte nella versione estesa di questo lavoro (Mecca, 2012).

3 La Classificazione delle Riviste

Utilizzando le tecniche discusse, è stata generata un'ampia classificazione di riviste. La classificazione discende dall'integrazione di 4 classificazioni internazionali ed 8 classificazioni nazionali:

- la classificazione basata sugli IF della ISI Wos (ISI);
- la classificazione basata sull'indice SJR di Scopus (Scopus);
- la classificazione australiana ERA (ERA);
- la classificazione ERIH della European Science Foundation (ESF ERIH);
- le classificazioni prodotto nell'ambito della VQR dei GEV delle Aree 01, 08, 09, 10, 11, 12, 13, 14.

La classificazione risultante è disponibile all'indirizzo http://db.unibas.it/valutazione.

4 La Classificazione dei Convegni

Il presupposto alla base della classificazione dei convegni è quello di valutare, nei casi in cui questo è opportuno, una pubblicazione apparsa negli atti di un convegno alla pari di una pubblicazione su rivista. Si tratta di una pratica non comune: in molte aree disciplinari le pubblicazioni in atti di convegno vengono considerati lavori di secondo piano rispetto alle riviste. In effetti, nei criteri fissati per la VQR (VQR), vari dei GEV dicono chiaramente che un lavoro in atti di convegno non può essere valuta-

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

to oltre la classe C o addirittura D. Più specificamente, questo è detto esplicitamente per l'Area 02, 03, 04, 05, 06 e 07. Relativamente all'Area 01 – che accomuna matematici ed informatici – è certamente vero per i matematici.
Di conseguenza, il problema della classificazione si pone principalmente per l'area informatica, le aree dell'ingegneria, e le aree umanistiche. La disponibilità di ranking nazionali ed internazionali in queste tre aree è molto disomogenea. Per cominciare, non esistono ranking di convegni di area umanistica. L'unico ranking noto di convegni di area ingegneristica è il ranking australiano (ERA); nel ranking, sono classificati circa 300 convegni delle varie aree dell'ingegneria. Esistono, viceversa, numerosi ranking di convegni di ambito informatico, per i quali un'ampia letteratura (CSTB, 1994) (Patterson, 1999) (Meyer, 2009) (Rahm, 2005) (Fortnow, 2009) sottolinea la necessità di considerare gli atti dei convegni quali sedi pubblicazione di primo piano; non a caso, oltre 1500 dei convegni classificati dagli australiani sono di area informatica. Lo stesso GEV dell'Area 01, nelle Frequently Asked Questions, risponde a questo proposito: "Siamo consapevoli dell'importanza delle conferenze per la ricerca in informatica, in alcuni casi non inferiore a quella di riviste in classe di merito 1."
In questo contesto, dovendoci necessariamente dotare di una classificazione dei convegni per tenere adeguatamente in considerazione un aspetto centrale delle pratiche di pubblicazione di alcuni settori, pare opportuno da una parte limitare il ranking alle fasce più alte, quelle da cui è più ragionevole attendersi un livello scientifico comparabile a quello delle riviste; d'altra parte, è necessario adottare una strategia in cui i valori di ranking risultino sufficientemente autorevoli. In particolare, per l'area informatica, abbiamo considerato 6 diversi ranking di convegni informatici:

- il ranking australiano ERA (ERA), integrato con i convegni di fascia A+ del ranking Core (CORE);
- il ranking del sito Microsoft Academic Search (Microsoft Academic Search), limitandosi ai primi 1000 convegni;
- il ranking del sito Arnetminer (ArnetMiner, 2008), limitandosi ai primi 1000 convegni;
- il ranking dell'Università dell'Alberta (Zaiane);
- il ranking della Technical University of Singapore (NTU);
- il ranking del GRIN.

5 La Classificazione delle Case Editrici

Tra le attività di classificazione delle sedi di pubblicazione, quella delle case editrici è decisamente la meno consolidata. A livello internazionale esistono pochissimi ranking di ampio respiro. Per i nostri scopi, abbiamo utilizzato il ranking dell'organizzazione olandese SENSE (SENSE) e quello del Finnish Publication Forum (Finnish Publication Forum). Più in dettaglio:

- il ranking SENSE classifica circa 1200 case editrici nelle fasce A, B, C, D;
- il ranking finlandese, classifica anch'esso circa 1200 case editrici; per la classificazione parallela delle riviste, il ranking finlandese utilizza 3 fasce, 1, 2, e 3, secondo

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

la convenzione per cui 3 è la fascia migliore (cioè la fascia A) ed 1 la peggiore (cioè la fascia C); per le case editrici, vengono utilizzate esclusivamente le fasce 1 e 2; di conseguenza, abbiamo adottato la convenzione di associare alle case editrici di fascia 1 il valore di ranking C, ed a quelle di fascia 2 il valore di ranking B.

Nel complesso, dall'integrazione dei due ranking è possibile ottenere dati di classificazione per circa 2100 case editrici. Questo dato, però, non è confortante per due ragioni. Per cominciare, la sovrapposizione tra le due classificazioni è bassissima – solo 200 case editrici su 1200; questo significa che la classificazione integrata è ampia, ma relativamente poco solida, dal momento che la stragrande maggioranza dei valori di ranking si basa su un'unica sorgente. Inoltre, molte delle case editrici italiane di rilievo sono del tutto assenti.

Bibliografia

ArnetMiner. (2008). *Computer Science Conference Ranking*. Tratto da http://v1.arnetminer.org/page/conference-rank/html/All-in-one.html

Casati, F. a. (2009). *Is Peer Review any Good? A Quantitative Analysis of Peer Review.* LiquidPub Project Technical Report. http://eprints.biblio.unitn.it/1654/.

CORE. (s.d.). *Computing Research and Education Association of Australia - Conference Ranking*. Tratto da http://core.edu.au/index.php/categories/conference%20rankings/1

CSTB. (1994). *Computer Science and Telecommunications Board - Academic Careers for Experimental Computer Scientists and Engineers.* CSTB Report. http://www.nap.edu/openbook.php?record_id=2236.

ERA. (s.d.). *Excellence of Research Australia*. Tratto da http://www.arc.gov.au/era

ESF ERIH. (s.d.). *European Science Foundation - European Reference Index for the Humanities*. Tratto da https://www2.esf.org/asp/ERIH/Foreword/search.asp

Finnish Publication Forum. (s.d.). *Ranking delle Case Editrici*. Tratto da http://www.tsv.fi/julkaisufoorumi/english.html

Fortnow, L. (2009). Time for Computer Science to Grow-Up. *Communications of the ACM, 58*(8), 33-35.

GRIN. (2000). *Gruppo Nazionale di Informatica - Classificazione delle Conferenze*. Tratto da http://www.grin-informatica.it/opencms/opencms/grin/ricerca/valutazione/classificazione_grin.html

Iglesias, J. E. (2007). Scaling the h-Index for Different Scientific ISI Fields. *Scientometrics, 73*(3), 303-320.

ISI. (s.d.). *Thomson Reuters - ISI Web of Science*. Tratto da http://www.isiknowledge.com/WOS

Mecca, G. a. (2012). *Sulla Classificazione delle Sedi di Pubblicazione nella Valutazione della Produzione Scientifica.* Università della Basilicata, Dipartimento di Matematica e Informatica, http://www.db.unibas.it/valutazione.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Meyer, B. a. (2009). Research Evaluation for Computer Science. *Communications of the ACM, 52*(4), 31-34.

Microsoft Academic Search. (s.d.). *Computer Science Conference Ranking*. Tratto da http://academic.research.microsoft.com/?SearchDomain=2

Ministero dell'Università e della Ricerca. (2012). D.M. n. 19 del 27 gennaio 1012 - Valorizzazione dell'efficienza delle universita' e conseguente introduzione di meccanismi premiali nella distribuzione di risorse pubbliche sulla base di criteri definiti ex ante anche mediante la previsione di un sistema d.

NTU. (s.d.). *Nanyang Technical University of Singapore - Computer Science Conference Ranking*. Tratto da http://www.ntu.edu.sg/home/assourav/crank.htm

Patterson, D. a. (1999). Computing Research Association - Best Practices Memo: Evaluating Computer Scientists and Engineers for Promotion and Tenure. *Computing Research News, A-B*.

Rahm, E. a. (2005). Citation Analysis of Database Publications. *SIGMOD Record, 34*(4), 48-53.

Scopus. (s.d.). *Elsevier - Scopus*. Tratto da http://www.info.sciverse.com/scopus

SENSE. (s.d.). *SENSE Publisher Ranking*. Tratto da http://www.sense.nl/qualityassessment

SIDEA. (s.d.). *Società Italiana di Economia Agraria - Relazione di Sintesi del Gruppo di Valutazione della Ricerca*. Tratto da https://ilo.unimol.it/sidea/images/upload/comitati/relazione_di_sintesi_valutazione.doc

SIE. (s.d.). *Società Italiana degli Economisti - Classificazione delle Riviste*. Tratto da http://www.siecon.org/online/anvur-cepr-cun-documenti/classificazione-per-fasce-delle-riviste-di-economia/

Thompson, B. (1993). *GRE Percentile Ranks Cannot Be Added or Averaged: A Position Paper Exploring the Scaling Characteristics of Percentile Ranks, and the Ethical and Legal Culpabilities Created by Adding Percentile Ranks in Making "High-Stakes" Admission Decisions*. http://www.eric.ed.gov/ERICWebPortal/contentdelivery/servlet/ERICServlet?accno=ED363637: Education Resources Information Center Report ED363637.

VQR. (s.d.). *ANVUR - Valutazione della Qualità della Ricerca 2004-2010*. Tratto da http://www.anvur.org/?q=schema-dm-vqr-definitivo

Wikipedia - M. (s.d.). *Mediana*. Tratto da http://it.wikipedia.org/wiki/Mediana_(statistica)

Wikipedia - MF. (s.d.). *Wikipedia - Majority Function*. Tratto da http://en.wikipedia.org/wiki/Majority_function

Wikipedia - Moda. (s.d.). *Moda*. Tratto da Wikipedia: http://it.wikipedia.org/wiki/Mediana_(statistica)

Zaiane, O. (s.d.). Tratto da Computer Science Conference Ranking, University of Alberta: http://webdocs.cs.ualberta.ca/~zaiane/htmldocs/ConfRanking.html

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Discovering Hidden `me` Edges in a Social Internetworking Scenario

Francesco Buccafurri, Gianluca Lax, Antonino Nocera, and Domenico Ursino*

DIMET, Università "Mediterranea" di Reggio Calabria, Via Graziella, Località Feo di Vito, 89122 Reggio Calabria, Italy
{bucca,lax,a.nocera,ursino}@unirc.it

Abstract. In Social Internetworking analysis, bridge users play a key role when information crossing among different Online Social Networks (OSNs) is investigated. Unfortunately, not always users make explicit that they are bridges by specifying the so-called `me` edges (i.e., those edges connecting the accounts of the same user in distinct OSNs). Thus, discovering hidden `me` edges is an important problem to face in this context. In this paper we propose an effective approach giving good results in real life settings. Indeed, an experimental campaign shows that our approach returns precise and complete results.

Keywords: Online Social Networks, Social Internetworking Scenario, Link Mining, Bridge Users, `me` edges.

1 Introduction

In the last years Online Social Networks (OSNs, for short) have been showing an enormous development and have become one of the main actors of the Web 2.0. Many researchers from disparate fields started to investigate them [19]. In particular, many approaches which collect large amounts of data from an OSN and apply techniques of classical Social Network Analysis on them have been proposed [11]. They obtain numerous and extremely interesting results. Many of these approaches model an OSN as a graph since they are based on the intuition that there is a strong correspondence between the user behavior in an OSN and the structural properties of the corresponding graph. However, the current research on OSNs should consider that nowadays users tend to spread their activities among more OSNs and, often, to show a different behavior in different OSNs. Think, for instance, of a user who joins Facebook to communicate with her friends and LinkedIn to find a job. Therefore, different OSNs are interconnected with each other and form a global graph. This complex structure represents the substrate of the so called Social Internetworking Scenario (SIS, for short). In this scenario a user can join multiple OSNs, two users can interact with each other even though they joined two different OSNs and did not know each other: it is sufficient that their communication passes through a bridge user, i.e. a

* Corresponding author

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

user who created an account in both the OSNs. As a consequence, bridge users represent a key aspect of a SIS. The links connecting the different accounts of the same bridge user in different OSNs are called `me` edges and, clearly, play a key role in a SIS. Several users explicitly specify their different accounts in the different OSNs they joined, and, consequently, their `me` edges. In the last years this activity is also facilitated by some support tools. However, many users, due to disparate reasons, do not perform this specification, and yet the knowledge of these edges could be extremely useful not only to allow users of different OSNs to communicate with each other but also to better construct the complete profile of a user. This last feature could be significant in various application fields. For instance, in e-commerce it could help to better identify users needs and desires and to perform personalized offers; in e-recruitment it could help to better know the complete profile of a candidate taking into account not only the information officially declared by her in her curriculum vitae but also the one she informally expressed in the OSNs she joined.

In this paper, we aim at providing a contribution in this setting. In particular, we propose an approach to discovering hidden `me` edges in a SIS. Given two nodes representing two accounts of two different OSNs, such that an explicitly declared `me` edge between them does not exist, in order to discover it (if any), our approach examines both the similarity of the accounts corresponding to the two nodes and the node neighbors. In order to motivate the above choice it is important to clarify that the purpose of our approach does not concern the case in which a user voluntarily keeps separated two accounts in their respective OSNs. In this case the user chooses account names very different from each other and does not have common friends and, very probably, one of the two profiles is fake (i.e., it does not contain real information about the user). This situation, which is closer to identity-management and security problems, is little relevant for our context, where we are interested in completing the real profile of users. For these users we may expect (and this is just what we found for `me`-connected users) that a user belonging to two (or more) OSNs tend to have at least partially overlapping sets of friends in the different OSNs. Therefore, the neighbors are useful information to exploit in order to detect hidden `me` edges. In the literature, several string similarity functions have been proposed (e.g., Jaro-Winkler, Levenshtein, QGrams, Monge-Elkan, Soundex, etc. [14]). One of them could be adopted to measure account similarity in our approach. In order to verify which is the best one (i.e., the one leading our approach to obtain the best performances) we carried out an experimental analysis; this is described in detail in Section 4.1. In order to determine the neighborhood of a node we exploit the information about the corresponding user, declared by means of XFN (XHTML Friends Network) [2] and FOAF (Friend-Of-A-Friend) [1], two standards specifically conceived for encoding human relationships. It is worth pointing out that the technicalities concerning these two standards have not to be manually handled by the user. As a matter of fact, each OSN has suitable mechanisms to automatically manage them in a way transparent to her, who has just to specify her relationships in a friendly fashion. Since, in a SIS, the number

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

of node pairs to consider for the possible presence of a me edge is enormous, we have identified a mechanism leading our approach to consider only a reasonable number of very promising pairs. More specifically, from the examination of the explicitly declared me edges, we have found that with a high probability some of the nodes belonging to the neighbors of two nodes linked by a me edge are, in their turn, linked by a me edge. As a consequence, our approach starts from a set of already known me edges and examines only the neighbors of the nodes involved in these edges. The plan of this paper is as follows: in the next section we examine related literature. In Section 3 we provide a detailed description of the proposed approach. In Section 4 we illustrate the experiments we have carried out to verify its performance. Finally, in Section 5 we draw our conclusions.

2 Related Work

Our approach can be related to link prediction (or, equivalently, to missing link detection), since we may image that a user who is detected as belonging to two OSNs even though she did not declared the me edge, eventually will insert this edge. Link prediction is a task of link mining aiming at predicting the (even future) existence of a link between two objects [22,4]. In the contest of Social Networks, it focuses on predicting friendships among users. Often, OSNs are represented as graphs [8]; as a consequence, some link prediction approaches are totally based on the structural properties of the graph representing the OSN [21]. A first possibility to perform this task consists in analyzing common neighbors. For instance, the authors of [25] have found a correlation between the number of common neighbors of two nodes of a collaboration network and the probability that these last ones will collaborate in the future. In order to decide whether two nodes are related, [3] exploits a similarity measure derived from the Jaccard coefficient. Based on *preferential attachment* [26], [6] verifies experimentally that the probability of a relationship between two nodes is proportional to the product of the number of their neighbors. Some approaches to link prediction rely on the notion of shortest-path distance which is computed by means of several similarity measures, like the Katz coefficient [18], PageRank [7] and SimRank [16]. Due to the high computational cost of these measures, some approximations have to be adopted in order to make them effective. In any case, whenever the number of nodes is considerable, the application of these methods may result in a too long running time. In conjunction with all the above techniques, some strategies may be used to enhance the accuracy of predictions. For instance, low-rank approximation [23] receives the adjacency matrix of the graph associated with a Social Network and reduces the *noise* inside it, yet preserving its structural properties. Also the use of *unseen bigrams* [13] can help in the link detection task. Here, the similarity between a node A and a node B is computed by taking into account the similarity between the nodes B and C, where this last one is the node most similar to A. Furthermore, the quality of link detection can be improved by means of clustering techniques aiming at identifying the graph components which introduce noise in the similarity com-

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

putation [15,21]. [27] proposes the application of statistical relational learning to link prediction in the domain of scientific literature citations. In this approach statistical modeling and feature selection are integrated into a search mechanism over the space of database queries in such a way as to define feature candidates involving complex interactions among objects in a given relational database. [28] analyzes the localization in space and time of a large number of users by means of their call detail records. Its analysis shows that users with similar movement routines are strongly connected in an OSN and have intense direct interactions. This result allows implicit ties in the OSN to be predicted with a significant accuracy starting from the analysis of the correlation between user movements (i.e., their mobile homophily). Other approaches to link detection come from the fields of *deduplication* and *disambiguation*. In particular, [5] proposes an algorithm for discovering duplicates in the dimensional tables of a Data Warehouse. As far as disambiguation is concerned, the similarity between entities has been exploited in [17] to distinguish references in a relationship-based data cleaning system. This approach models a database as a graph of entities linked to each other by relationships and exploits both node features and relationships to carry out the disambiguation task. From the above analysis it emerges that our approach, among the above literature, can be related only with common-neighbors approaches. However, despite the apparent closeness to them, we can easily realize that they are not directly applicable to our context. Indeed, the notion of common-neighbors relies, in general, on the notion of common identity of friends of a user. But discovering the common identity of users (in different OSNs) is for us the output of the problem, leading to a sort of recursive definition of the problem itself. We have confirmed experimentally the above claim by showing that the application of the state-of-the-art common-neighbors approaches to our problem returns very unsatisfying results. For space reasons, we do not include these experiments in this paper, but the reader can find them in [9].

3 Description of the proposed approach

In this section we illustrate in detail our approach for discovering hidden `me` edges in a SIS. It consists of three functions that will be examined in the following.

Function findMeEdges. It receives a set *startmeSet* of existing `me` edges and returns a set *newmeSet* of discovered `me` edges. For each edge e of *startmeSet*, our approach considers the involved nodes n_a and n_b. Then, it computes their neighborhoods $neigh_a$ and $neigh_b$. In our approach the neighborhood of a node consists of those nodes linked to it by a `contact` or a `friend` relationship in XFN or FOAF. After this, our approach considers all the possible pairs (n'_a, n'_b) such that $n'_a \in neigh_a$ and $n'_b \in neigh_b$. For each of these pairs it first computes the similarity between the account names of n'_a and n'_b by calling a function $accountSim(n'_a, n'_b)$. As pointed out in the Introduction, there exist several already defined functions for computing the similarity between two strings, each characterized by specific features (e.g., Jaro-Winkler, Levenshtein, QGrams,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Monge-Elkan, Soundex, etc. [14]); $accountSim(n'_a, n'_b)$ can exploit one of these functions on the basis of the desired severity level; for instance, QGrams is very severe and assigns quite low similarity degrees; Jaro-Winkler is more permissive whereas Soundex is very permissive. In Section 4.1 we shall discuss about the exploitation of these functions in our approach. If the similarity values detected by $accountSim(n'_a, n'_b)$ is higher than a certain threshold th_{cand} and there does not already exist an explicitly declared me edge between n'_a and n'_b, our approach verifies if a hidden me edge exists between these two nodes. For this purpose it applies a function sim (which we shall explain in detail below) on n'_a and n'_b. If the result of this function is higher than a certain threshold th_{real}, our approach assumes that a hidden me edge exists between n'_a and n'_b and inserts it in $newmeSet$. In the opposite case, our approach assumes that no me edge exists between n'_a and n'_b. The algorithm implementing our approach is described in Algorithm 1.

As for the computational complexity of this algorithm, it is possible to state the following theorem:

Theorem 1. *Given a pair of nodes n_a and n_b, the computational complexity for inferring a hidden* me *edge between them is $O(d^2)$, where $d = max(|neigh_a|, |neigh_b|)$, being $neigh_a$ and $neigh_b$ the neighborhoods of n_a and n_b, resp.* □

Function sim. This function receives: *(i)* a positive integer k; *(ii)* a list $simList$ of triplets $\langle node_a, node_b, \overline{sim_{ab}} \rangle$ each representing a pair of nodes along with their "basic" similarity value; this last one coincides with $accountSim(node_a, node_b)$ when $k = 1$; when $k > 1$ it is obtained by suitably weighting $accountSim(node_a, node_b)$ on the basis of the degrees of $node_a$ and $node_b$ (see, below, Algorithm 2); *(iii)* a real number $pSim_{ab}$ representing the similarity value of n'_a and n'_b at step $k - 1$. It returns the similarity sim_{ab} between n'_a and n'_b at step k. It behaves as follows: if $k = 1$ then sim_{ab} coincides with $pSim_{ab}$. Otherwise, sim computes the average value $avgSim$ of the similarities of the node pairs of $simList$ and, then, computes sim_{ab} as a weighted mean of $pSim_{ab}$ and $avgSim$. The weight w assigned to $avgSim$ is set to $\frac{1}{k^k}$ in such a way that it quickly decreases when k increases. This implies that the nodes belonging to the closest neighbors of n'_a and n'_b provide a much higher contribution than the ones belonging to the farthest neighbors of n'_a and n'_b. If the absolute value of the difference between sim_{ab} and $pSim_{ab}$ is less than a certain threshold ϵ then sim terminates and returns sim_{ab}. Otherwise, for each triplet $\langle node_a, node_b, \overline{sim_{ab}} \rangle$ in $simList$, it inserts in a list $nextSimList$ the triplets $\langle node'_a, node'_b, \overline{sim'_{ab}} \rangle$ such that: *(i)* $node'_a$ belongs to the neighborhood of $node_a$; *(ii)* $node'_b$ belongs to the neighborhood of $node_b$; *(iii)* $\overline{sim'_{ab}}$ represents the basic similarity of $node'_a$ and $node'_b$. For each triplet $\langle node_a, node_b, \overline{sim_{ab}} \rangle$ the corresponding triplets $\langle node'_a, node'_b, \overline{sim'_{ab}} \rangle$ inserted in $nextSimList$ are chosen among those ones having the highest values of $\overline{sim'_{ab}}$. This task is performed by the function $functionMaxs$ which will be explained in detail below. After $nextSimList$ has been constructed, sim invokes recursively itself by passing $k + 1$, $nextSimList$ and sim_{ab}. The algorithm implementing sim is described in Algorithm 2.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

Algorithm 1 $findMeEdges$

Require: We denote by $accountSim(node_a, node_b)$ a function returning the similarity value between the account names of $node_a$ and $node_b$, by $existMe(node_a, node_b)$ a boolean function returning true if a `me` edge between $node_a$ and $node_b$ exists and false otherwise, and by $sim(step, simList, pSim_{ab})$ a function computing the similarity between $node_a$ and $node_b$ on the basis of the similarity of the corresponding neighbors.
Input $startmeSet$: a set of existing me edges
Output $newmeSet$: a set of discovered me edges
Constant th_{cand} {A threshold for candidate `me` edges}
Constant th_{real} {A threshold for real discovered `me` edges}
Variable e: an edge
Variable n_a, n_b, n'_a, n'_b: a node
Variable $\overline{sim_{ab}}$: a support variable
Variable $neigh_a, neigh_b$: a list of nodes
Variable $simList$: a list of triplets of the form $\langle node_a, node_b, \overline{sim_{ab}} \rangle$

```
 1: simList := ∅; newmeSet := ∅; neigh_a := ∅; neigh_b := ∅
 2: for i := 1 to |startmeSet| do
 3:    extract an edge e from startmeSet
 4:    get the nodes n_a and n_b of e
 5:    insert the neighbors of n_a in neigh_a
 6:    insert the neighbors of n_b in neigh_b
 7:    for each node n'_a in neigh_a do
 8:       for each node n'_b in neigh_b do
 9:          sim_ab:=accountSim(n'_a, n'_b)
10:          if (sim_ab > th_cand) and !existMe(n'_a, n'_b) then
11:             insert into simList the triplet ⟨n'_a, n'_b, sim_ab⟩
12:             if (sim(1, simList, sim_ab) > th_real) then
13:                insert the me edge between n'_a and n'_b in newmeSet
14:             end if
15:             simList := ∅
16:          end if
17:       end for
18:    end for
19: end for
20: return newmeSet
```

Observe row 3 of Algorithm 2; cur_max_num, which represents the real number of node pairs to consider at step k of the computation of sim_{ab}, is obtained by multiplying max_num (which represents the maximum number of node pairs to consider in the computation of sim_{ab}) by w. Due to the definition of w, this implies that cur_max_num quickly decreases when k increases; therefore, when k increases, the number of pairs which can contribute to sim_{ab} highly decreases.

Function findMaxs. The function $findMaxs$ receives two nodes $node_a$ and $node_b$ and a positive integer cur_max_num. It returns a list $simList'$ of cur_max_num triplets $\langle node'_a, node'_b, \overline{sim'_{ab}} \rangle$. It behaves as follows: first it constructs the neighborhoods $neigh_a$ of $node_a$ and $neigh_b$ of $node_b$. After this, for each pair $\langle node'_a, node'_b \rangle$ such that $node'_a \in neigh_a$ and $node'_b \in neigh_b$, it computes the corresponding similarity $scaledSim_{ab}$. This last one considers the similarity $accountSim(node'_a, node'_b)$ (see Algorithm 1), as well as the degrees of $node_a$, $node'_a$, $node_b$ and $node'_b$. This choice aims at avoiding that $node'_a$ and $node'_b$ are power users and $node_a$ and $node_b$ are not; in this case, it could happen that $node'_a$ (resp., $node'_b$) belongs to $neigh_a$ (resp., $neigh_b$) only because it corresponds to a very famous person (think, for instance, to the case $node'_a$ and $node'_b$ represent 'Barack Obama'); in this case the presence of this node in $neigh_a$ and $neigh_b$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

Algorithm 2 *sim*

Require: We denote by $max(x, y)$ a function returning the maximum value between x and y, by $findMaxs(node_a, node_b, cur\_max\_num)$ a function returning a list of triplets $\langle node'_a, node'_b, \overline{sim_{ab}}' \rangle$ such that $\langle node'_a, node'_b \rangle$ is one of the cur_max_num pairs of nodes having the highest similarity value $\overline{sim_{ab}}'$ among all the possibile pairs of nodes obtained by taking a node of the neighborhood of $node_a$ and a node of the neighborhood of $node_b$.
Input k: the current step
Input $simList$: a list of triplets $\langle node_a, node_b, \overline{sim_{ab}} \rangle$ each representing a pair of nodes along with their basic similarity value
Input $pSim_{ab}$: the similarity value of n'_a and n'_b at step $k-1$
Output sim_{ab}: the similarity value of n'_a and n'_b at step k
Constant max_num {The maximum number of node pairs to consider in the computation of sim_{ab}}
Constant ε {A parameter determining the algorithm termination}
Variable w: a weight in the real interval [0,1]
Variable cur_max_num: the current number of node pairs having the highest similarity values to consider
Variable $avgSim$: a real variable
Variable $nextSimList$: a new list of triplets $\langle node_a, node_b, \overline{sim_{ab}} \rangle$ whose structure is analogous to the one of $simList$

1: $nextSimList := \emptyset$;
2: $w := \frac{1}{k^k}$
3: $cur\_max\_num := max(\lceil max\_num \cdot w \rceil, 1)$
4: **if** (k=1) **then**
5: $sim_{ab} := pSim_{ab}$
6: **else**
7: assign to $avgSim$ the average value of the similarities of the node pairs of $simList$
8: $sim_{ab} := (1-w) \cdot pSim_{ab} + w \cdot avgSim$
9: **end if**
10: **if** $(|sim_{ab} - pSim_{ab}| \geq \varepsilon)$ **then**
11: **for each** $\langle node_a, node_b, \overline{sim_{ab}} \rangle$ in $simList$ **do**
12: insert the list returned by $findMaxs(node_a, node_b, cur\_max\_num)$ in $nextSimList$
13: **end for**
14: **return** $sim(k+1, nextSimList, sim_{ab})$
15: **else**
16: **return** sim_{ab}
17: **end if**

is not significant in defining the real life relationships of $node_a$ and $node_b$. More specifically, $scaledSim_{ab}$ is computed as:

$$scaledSim_{ab} := min\left(\frac{max(D(node'_a), D(node_a))}{max(D(node'_a), D(node_a)) + |D(node'_a) - D(node_a)|}, \frac{max(D(node'_b), D(node_b))}{max(D(node'_b), D(node_b)) + |D(node'_b) - D(node_b)|}\right) \cdot accountSim(node'_a, node'_b)$$

Here, $D(node_x)$ returns the degree of $node_x$. $findMaxs$ inserts in $simList'$ the cur_max_num triplets $\langle node'_a, node'_b, scaledSim_{ab} \rangle$ having the highest values of $scaledSim_{ab}$. Finally, it returns $simList'$. The algorithm implementing $findMaxs$ is described in Algorithm 3.

4 Experiments

In this section we present our experimental campaign conceived to determine the performances of our approach. Since this last one operates on a SIS and not on an OSN, we had to extract not only the connections among the accounts of

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

Algorithm 3 $findMaxs$

Require: We denote by $accountSim(node_a, node_b)$ a function returning the similarity value between the account names of $node_a$ and $node_b$, by $max(x, y)$ the function returning the maximum value between x and y, by $min(x, y)$ the function returning the minimum value between x and y and by $D(node_x)$ the function returning the degree of $node_x$.
Input $node_a, node_b$: a node
Input cur_max_num: the number of node pairs to return
Output $simList'$: a list of cur_max_num triplets of the form $\langle node'_a, node'_b, \overline{sim'_{ab}} \rangle$
Variable $node_a, node_b, node'_a, node'_b$: a node
Variable $neigh_a, neigh_b$: a list of nodes
Variable t'_{ab}: a triplet of the form $\langle node'_a, node'_b, \overline{sim'_{ab}} \rangle$
Variable $cont, scaledSim_{ab}$: an integer variable
Variable df_a, df_b: a real variable

1: $simList' = \emptyset$; $neigh_a = \emptyset$; $neigh_b = \emptyset$
2: insert the neighbors of $node_a$ in $neigh_a$
3: insert the neighbors of $node_b$ in $neigh_b$
4: $cont := 0$
5: **for each** $node'_a$ in $neigh_a$ **do**
6: **for each** $node'_b$ in $neigh_b$ **do**
7: $df_a := \frac{max(D(node'_a), D(node_a))}{max(D(node'_a), D(node_a)) + |D(node'_a) - D(node_a)|}$
8: $df_b := \frac{max(D(node'_b), D(node_b))}{max(D(node'_b), D(node_b)) + |D(node'_b) - D(node_b)|}$
9: $scaledSim_{ab} := min(df_a, df_b) * accountSim(node'_a, node'_b)$
10: **if** $(cont < cur\_max\_num)$ **then**
11: insert in $simList'$ the triplet $\langle node'_a, node'_b, scaledSim_{ab} \rangle$
12: sort $simList'$ in a descending order
13: $cont++$
14: **else**
15: $t'_{ab} := simList'.get(cur\_max\_num - 1)$
16: **if** $(scaledSim_{ab} > t'_{ab}.\overline{sim'_{ab}})$ **then**
17: replace the triplet in the position $(cur\_max\_num - 1)$ of $simList'$ with the triplet $\langle node'_a, node'_b, scaledSim_{ab} \rangle$
18: sort $simList'$ in a descending order
19: **end if**
20: **end if**
21: **end for**
22: **end for**
23: **return** $simList'$

different users in the same OSN but also the connections among the accounts of the same user in different OSNs.

In order to represent these connections, two standards encoding human relationships are generally exploited. The former is XFN (XHTML Friends Network) [2]. It simply uses an attribute, called `rel`, to specify the kind of relationship between two users. Some possible values of `rel` are `friend`, `contact`, `co-worker`, `parent`, and so on. A (presumably) more complex alternative to XFN is FOAF (Friend-Of-A-Friend) [1]. In our experiments, we considered a SIS consisting of four OSNs, namely Twitter, LiveJournal, YouTube and Flickr. We chose these four OSNs because they were largely analyzed in the past in Social Network Analysis papers devoted to study a single OSN or to compare different OSNs [20,24,12,29]. For our experiments, we exploited a server equipped with a 2 Quad-Core E5440 processor and 16 GB of RAM with the CentOS 6.0 Server operating system. We performed our experiments from January 30, 2012 to March 08, 2012. Exploited data can be found at the URL address `http://www.ursino.unirc.it/sebd-12.html`.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

4.1 Unsupervised method validation

A first experiment aimed at determining the performance of our approach as well as at choosing the best function for computing the account name similarity in our context. For this purpose we performed a pre-processing task consisting of two steps. Specifically, during the first step we constructed a set *meSet* of 100 node pairs such that a `me` edge existed between them; this was trivial since it was sufficient to find `me` edges declared in XFN and FOAF. During the second step we detected a set *notmeSet* of 100 node pairs such that a me edge did not exist between them; for this purpose, we detected a `me` edge (n_a, n_b) and we obtained an element of *notmeSet* by taking the pair (n_c, n_b) such that there existed a `contact` or a `friend` edge between n_a and n_c. Then, we applied our approach on the pairs of *meSet* and *notmeSet* and we obtained a set $meSet'$ (resp., $meSet''$) of pairs of *meSet* (resp., *notmeSet*) for which it detected a `me` edge, as well as a set $notmeSet'$ of *meSet* of pairs for which it did not detect a `me` edge. It is worth pointing out that in this experiment we modified the input and the output of our approach in such a way that it simply receives two nodes and returns *true* if it detects a `me` edge between them, *false* otherwise. To compute the performance of our approach we adopted two classical measures, namely *Precision* and *Recall*. In the literature Precision is an indicator of correctness, whereas Recall is an indicator of completeness. In our case they were defined as follows: $Precision = \frac{|meSet'|}{|meSet'|+|meSet''|}$; $Recall = \frac{|meSet'|}{|meSet'|+|notmeSet'|}$.

In this formulas we assigned the same importance to the approach's capabilities of detecting `me` and not `me` edges. Actually, the behavior of our approach (and, consequently, the values of Precision and Recall) depends on the function adopted for computing the account name similarity. As a consequence, we considered the most common of these functions and, for each of them, we computed the Precision and the Recall of our approach. In this way we were able to determine the function maximizing these measures. Obtained results are shown in Table 1.

Function	*Precision*	*Recall*
Jaro-Winkler	0.558	0.920
QGrams	0.908	0.690
Levenshtein	0.877	0.710
Smith-Waterman	0.840	0.790
Smith-Waterman-Gotoh	0.779	0.810
Monge-Elkan	0.779	0.810
Needleman-Wunch	0.500	1.000
Jaro	0.555	0.910
Soundex	0.500	0.990

Table 1. Precision and Recall of our approach for each account name similarity function

From the analysis of this table we can observe that, *in our application scenario*, many functions are well suited for measuring account name similarity. Indeed, 5 functions led our approach to obtain a Precision higher than 0.77 and 6 functions led it to obtain a Recall higher than 0.81. These results show also that our approach presents a very satisfying performance both in correctness and in completeness. Finally, among the considered functions, QGrams (resp.,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Needleman-Wunch) proved to be the one capable of assuring the best Precision (resp., Recall). As for the computation of string similarities, our approach needed an average time of $10^{-5}s$ to process two nodes. The number of iterations ranged between 2 and 3; as a matter of fact, at the fourth iteration the value of w was 1/256, which makes the contribution of the fourth iteration negligible.

4.2 Supervised method validation

This experiment aimed at computing the correctness of our approach in a way different from the one considered in the previous experiment; in particular, in this case, we wanted to benefit from the support of a human expert. In this case we first applied a crawling technique to derive a sample of the SIS. This sample was necessary to have a starting set of `me` edges at disposal. In order to maximize the number of `me` edges present in the sample we applied BDS, a crawling technique specifically conceived to operate on a SIS instead of on a single OSN, which is highly capable of finding `me` edges [10]. Our sample consisted of 93169 nodes and 146325 edges, 745 of which were `me` ones. We randomly selected 160 `me` edges and put them in a set *startmeSet*. We gave this set in input to our approach. The adopted account name similarity function was QGrams because it proved to assure the best Precision. Our approach returned a set $finalmeSet$ of 22 `me` edges and a set $fullnotmeSet$ of 133 not `me` edges. From this last set we randomly selected a set $finalnotmeSet$ of 22 not `me` edges in such a way that `me` edges and not `me` edges had the same weight in the computation. After this, we required the human expert to manually verify if the elements of $finalmeSet$ represented real `me` edges and the elements of $finalnotmeSet$ represented real not `me` edges. For this purpose, she really visited the pages corresponding to the nodes of each edge. For each edge her possible answers were *true*, *false* and *unknown*. She gave this last answer when she was not able to access the page associated with a node or to give an answer with an absolute certainty. At the end of the experiment we obtained that, as for $finalmeSet$, she returned $t' = 16$ *true*, $f' = 4$ *false* and $u' = 2$ *unknown*. As for $finalnotmeSet$, she returned $t'' = 18$ *true*, $f'' = 2$ *false* and $u'' = 2$ *unknown*. After this, we computed the correctness of our approach by exploiting a metric well known in the literature, i.e. *accuracy*. It is defined as:

$$accuracy = \frac{t'+f'}{t'+f'+t''+f''} \cdot \left(\frac{t'}{t'+f'}\right) + \frac{t''+f''}{t'+f'+t''+f''} \cdot \left(\frac{t''}{t''+f''}\right) = \frac{t'+t''}{t'+f'+t''+f''} = 0.85$$

At the end of this experiment we can conclude that our approach really shows a very satisfying value of correctness in both an unsupervised and a supervised validation.

5 Conclusions

In this paper we have proposed an approach for discovering hidden `me` edges in a Social Internetworking Scenario. First, we have seen the motivations underlying our approach and its possible benefits. Then, we have examined related

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

literature. Afterwards, we have provided a detailed (both informal and formal) description of it. Finally, we have illustrated an experimental campaign devoted to determine its performances. SIS analysis is a very young research field and we plan to perform further research efforts in this context in the future. A first effort will be the development of some optimization functionalities to reduce the number of string matching operations required for each pair of accounts. A possible development could regard the definition of an approach that exploits both explicitly declared and discovered `me` edges to construct a very rich user profile expressing her needs, desires and behavior. This idea can be further developed in such a way as to find malicious users who create multiple accounts for frauds and other misbehaving activities and clearly do not explicitly declare the corresponding `me` edges. Our approach first could discover hidden `me` edges and then could determine that the behavior of the corresponding nodes is malicious. Finally, it could be possible to develop enhanced versions of already existing crawling techniques specifically conceived for SIS's instead of for OSNs and highly benefiting of `me` edges.

Acknowledgment
This work was partially funded by the Italian Ministry of Research through the PRIN Project EASE (Entity Aware Search Engines).

References

1. The Friend of a Friend (FOAF) project. `http://www.foaf-project.org/`, 2012.
2. XFN - XHTML Friends Network. `http://gmpg.org/xfn`, 2012.
3. L. Adamic and E. Adar. Friends and Neighbors on the Web. *Social Networks*, 25(3):211–230, 2003.
4. M. Al Hasan and M.J. Zaki. A Survey of Link Prediction in Social Networks. In *Social Network Data Analytics*, pages 243–276. Elsevier, 2011.
5. R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in Data Warehouses. In *Proc. of the International Conference on Very Large Data Bases (VLDB '02)*, pages 586–597, Hong Kong, China, 2002. VLDB Endowment.
6. A.L. Barabási, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3-4):590–614, 2002.
7. S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*, 30(1-7):107–117, 1998.
8. M. Brocheler, A. Pugliese, and V.S. Subrahmanian. Probabilistic Subgraph Matching on Huge Social Networks. In *Proc. of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM'11)*, pages 271–278, Kahosiung, Taiwan, 2011.
9. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Mining Links among Social Networks. *Technical Report. Available from the Authors*, 2012.
10. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Crawling Social Internetworking Systems. In *Proc. of the International Conference on Advances in Social Analysis and Mining (ASONAM 2012)*, Istanbul, Turkey, Forthcoming.
11. P. Carrington, J. Scott, and S. Wasserman. *Models and Methods in Social Network Analysis*. Cambridge University Press, 2005.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

12. X. Cheng, C. Dale, and J. Liu. Statistics and Social Network of Youtube Videos. In *Proc. of the International Workshop on Quality of Service (IWQoS 2008)*, pages 229–238, Enschede, The Netherlands, 2008. IEEE.
13. I. Dagan, F. Pereira, and L. Lee. Similarity-based estimation of word cooccurrence probabilities. In *Proc. of the annual meeting on Association for Computational Linguistics*, pages 272–278. Association for Computational Linguistics, 1994.
14. A.K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios. Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16, 2007.
15. D. Foster and L. Ungar. A proposal for learning by ontological leaps. In *Proc. of Snowbird Learning Conference, Snowbird*, 2002.
16. G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'02)*, pages 538–543, Edmonton, Alberta, Canada, 2002. ACM Press.
17. D.V. Kalashnikov and S. Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems*, 31(2):716–767, 2006.
18. L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
19. J. Kleinberg. The convergence of social and technological networks. *Communications of the ACM*, 51(11):66–72, 2008.
20. B. Krishnamurthy, P. Gill, and M. Arlitt. A few chirps about Twitter. In *Proc. of the First Workshop on Online Social Networks*, pages 19–24, Seattle, Washington, USA, 2008.
21. D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.
22. L. Lü and T. Zhou. Link Prediction in Complex Networks: A Survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.
23. C.D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press Cambridge, 2008.
24. A. Mislove, H.S. Koppula, K.P. Gummadi, F. Druschel, and B. Bhattacharjee. Growth of the Flickr Social Network. In *Proc. of the International Workshop on Online Social Networks (WOSN08)*, pages 25–30, Seattle, Washington, USA, 2008. ACM.
25. M.E.J. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.
26. M.E.J. Newman. The structure and function of complex networks. *SIAM review*, pages 167–256, 2003.
27. A. Popescul and L.H. Ungar. Statistical relational learning for link prediction. In *Proc. of the International Workshop on Learning Statistical Models from Relational Data*, volume 149, pages 172–179, Acapulco, Mexico, 2003.
28. D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.L. Barabási. Human mobility, social ties, and link prediction. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*, pages 1100–1108, San Diego, California, USA, 2011. ACM.
29. S. Ye, J. Lang, and F. Wu. Crawling online social graphs. In *Proc. of the International Asia-Pacific Web Conference (APWeb'10)*, pages 236–242, Busan, Korea, 2010. IEEE.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Count Constraints for Inverse OLAP and Aggregate Data Exchange*

Domenico Saccà, Edoardo Serra, and Antonella Guzzo

DEIS, Università della Calabria, 87036 Rende, Italy
{sacca,eserra,guzzo}@deis.unical.it

Abstract. A classical problem in database theory is to verify whether there exists a relation (or database) instance satisfying a number of given dependency constraints but the issue of handling constraints on aggregate data has not been much investigated so far. This paper introduces a new type of data dependency, called count constraint, that requires the results of given count operations on a relation to be within a certain range. Count constraints are defined by a suitable extension of first order predicate calculus, based on set terms, and they are then used in a new decisional problem, the Inverse OLAP: given a fact table, does there exist an instance satisfying a set of given count constraints? Count constraints can also be used into a data exchange system context, where data from the source database are transferred to the target database using aggregate operations.

Keywords: Count Constraints, OLAP Analysis, Data Exchange

1 Introduction

A typical problem in relational database theory is to decide the existence of a database satisfying a given set of given integrity constraints. Classical approaches mainly focus on inclusion dependencies and functional dependencies [2–4]. This problem has recently received a renewed deal of interest within the context of data exchange [5–7], but the issue of handling constraints on aggregate data has not been much investigated so far, notwithstanding the relevance of aggregate operations in many applications.

In this paper we consider a new type of data dependencies, called *count constraint*, prescribing the results of given count operations on a relation to be within a certain ranges. Count constraints are relevant in *OLAP analysis*, which is characterized by multidimensional data cubes that enable manipulation and analysis of data stored in a source database from multiple perspectives in a fast way [8, 9]. In this paper we apply count constraints to a *fact table*, that is a relation scheme whose attributes are dimensions (i.e., properties, possibly structured at various levels of abstraction) and measures (i.e., values computed on the basis of some aggregation operations, e.g., count). A fact table is part of a *star schema* that typically includes also dimension tables describing dimension attributes.

* This paper is an extended abstract of [1].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

To get an intuition of our approach, consider a fact table $\mathcal{R}(T, I)$ with two attributes T (the ID of a transaction) and I (the ID of an item) with domain $\mathcal{I}$, stored in a relation $\mathcal{D}_I$. Given a relation r on $\mathcal{R}$, $G = Group(r)By(T)$ divides r into a number of groups, one for each transaction ID. We want to express the following constraints: every item may occur in at most 1000 transactions, the itemset $i = \{a, b, c\}$ must occur as a transaction group at least 100 and at most 200 times, whereas every other set s of items cannot be present as a sub-group of more than 10 transactions, except for all subsets of i (including i) that have no limits. Such conditions can be formulated by the following count constraints, expressed with a logic formalism that extends the one adopted in the data exchange setting [5–7] – note that we write below i to represent the constant set term:

$$\forall \mathtt{I}\ (\ \mathcal{D}_{\mathtt{I}}(\mathtt{I})\ \rightarrow\ 0 \leq \#(\{\mathtt{T} : \mathcal{R}(\mathtt{T}, \mathtt{I})\}) \leq 1000\ ); \tag{1}$$

$$\forall \mathtt{s}\ (\ \mathtt{s} = i\ \rightarrow\ 100 \leq \#(\{\mathtt{T} : \mathtt{s} = \{\mathtt{I} :\ \mathcal{R}(\mathtt{T}, \mathtt{I})\}\}) \leq 200\ ); \tag{2}$$

$$\forall \mathtt{s}\,(\,\mathtt{s} \subseteq \{\mathtt{I} : \mathcal{D}_{\mathtt{I}}(\mathtt{I})\} \wedge \mathtt{s} \not\subseteq i \rightarrow 0 \leq \#(\{\mathtt{T} : \mathtt{s} \subseteq \{\mathtt{I} : \mathcal{R}(\mathtt{T}, \mathtt{I})\}\}) \leq 10\,). \tag{3}$$

Count constraints represents an extension of cardinality constraints that have been first introduced in the context of the entity-relationship model and have recently received a renewed interest. A declarative format to express them has been recently proposed in [10] with the aim of formulating more general characteristics in the process of data generation. Cardinality constraints have been also introduced within the formalism of logic programs by [11, 12]. Observe that, while constraint (1) can be easily expressed as a cardinality constraint, the other two cannot as they involve sets generated by complex grouping operations.

We use count constraints to define a new decisional problem, the *Inverse OLAP*: given a star schema consisting of one fact table, does there exist a relation instance satisfying a set of given count constraints? This problem extends the *Inverse Frequent itemset Mining* problem (`IFM` for short) [13–15] and has a potential high relevance in the contexts of generating synthetic data cubes having the same characteristics of real-world ones in terms of aggregation patterns and of privacy-preserving aggregate data exchange.

In[16, 1] we proved that the new problem is NEXP-complete under various conditions: data complexity (i.e, the number of attributes and the size of constraints are constant), program complexity (i.e, the domains are constant) and combined complexity. It is interesting to point out that the NEXP-complete results on the complexity of cardinality constraints have been detected both in [10] and in [12].

In the paper, we illustrate how count constraints can be also used in the context of data exchange. Traditionally the mapping of the data from the source to the target schema is defined by source- to-target TGDs (Tuple Generating Dependencies) and additional constraints on the target schema are specified in form of EGDs (Equality Generating Dependencies) and TGDs. We show that count constraints represent a powerful extension of EGDs.

The paper is organized as follows. In Section 2 we present the logic language for describing count constraints, introduce the inverse OLAP problem and discuss its complexity. We illustrate the usage of count constraints and the relevance of inverse OLAP in a motivating example in Section 3. In Section 4 we show how count constraints can be used into a data exchange system context. Finally in Section 5 we draw the conclusion and discuss further work.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2 Count Constraints

Let $U = (A_1, \ldots, A_n)$ be a list of n distinct attributes on the domains $D_1, \ldots, D_n$ with given cardinalities $d_1, \ldots, d_n$. A *relation scheme* is a pair consisting of a relation name and a list of attributes. In this paper we shall only deal with exactly one relation scheme containing all attributes in U, say $\mathcal{R}(A_1, \ldots, A_n)$. We also assume that the domains of the attributes $D_1, \ldots, D_n$ are stored in suitable tables of mono-attribute relation schemes — we denote their relation schemes by $\mathcal{D}_1(A_1), \ldots, \mathcal{D}_n(A_n)$. A *relation* on $\mathcal{R}$ (also called an *instance* of $\mathcal{R}$) is any table on U.

$\mathcal{R}$ represents a *star schema* consisting of a unique *fact table* whose attributes represent dimensions - as we are only interested in count aggregation, we omit to include measures. Some of the dimensions could be organized in layers defined by Functional Dependencies (FDs) — for instance the FDs $A \rightarrow B$ and $B \rightarrow C$ state that the values of dimension A are grouped at a first level B and at a second level C. In correspondence of FDs we may have additional domain relations (usually called *dimension tables*) describing hierarchies among two dimensions, e.g., $\mathcal{D}_{A,B}$ and $D_{B,C}$.

We next introduce an extension of first order predicate calculus to define count constraints on the instances of $\mathcal{R}$. The predicate symbols are: $\mathcal{R}$, the domain relation schemes $\mathcal{D}_1, \ldots, \mathcal{D}_n$ and possible dimension hierarchy domains. The *constants* of the language are the domain values (*domain constants*) and all (non-negative) integers.

Besides to domain constants, the Herbrand universe includes constant set terms – a set term represents a set of tuples having arity bound by some constant k. For instance, given the attributes A and B with domains $\{a_1, a_2, a_3\}$ and $\{b_1, b_2\}$ respectively, examples of constant set terms on $\{A, B\}$ are $\{[a_1, b_1], [a_2, b_1], [a_3, b_2]\}$ and $\{[a_2, b_1]\}$, while $\{[a_1], [a_3]\}$ and $\{[a_2]\}$ are two constant set terms on $\{A\}$.

A non-constant set term is defined as $\{x_1, \ldots, x_s : \alpha\}$, where $x_1, \ldots, x_s$ are variables and α is a *count formula* (defined next), in which $x_1, \ldots, x_s$ occur as free variables (similar notation for set terms and aggregate predicates has been used in the dlv system [11]). There is an interpreted function symbol $count$ (denoted by $\#$) that can be applied to a set term T to return the number of tuples in T (i.e., the cardinality of the table represented by T).

Our language is equipped with a countable number of variables and makes use of the following types of terms: (i) *simple term* (either a domain constant or a variable), (ii) *set term* (either a constant set term or a formula term) and (iii) an *integer term* (either an integer or a $count$ function term).

An *atom* can be: (i) a *relation predicate*, (ii) a *domain predicate* or (iii) a *comparison predicate* (equality or disequality of two terms, comparison of two integer terms, and inclusion of two set terms).

A *count constraint* C is a formula of one of the following two types:

1. $\forall \mathbf{X}\ (\ \phi(\mathbf{X}) \ \rightarrow\ \beta_{min} \leq \#(\{\ \mathbf{Y} : \exists \mathbf{Z}\ \psi(\mathbf{X}, \mathbf{Y}, \mathbf{Z})\ \}) \leq \beta_{max}\ )$ (*tuple count constraint*), or
2. $\forall \mathbf{X}\ (\ \phi(\mathbf{X}) \ \rightarrow\ \beta_{min} \leq \#(\ \{\ \mathbf{W} : t * \{\ \mathbf{Y} : \exists \mathbf{Z}\ \psi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W})\ \}\ \}) \leq \beta_{max}\ )$ (*group count constraint*),

where $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ and $\mathbf{W}$ are disjunct lists of variables ($\mathbf{X}$ and $\mathbf{Z}$ can be empty), ϕ is a (possibly empty) conjunction of domain and comparison predicates, ψ is a conjunction

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

of relation and comparison predicates, β_{min} and β_{max} are integers for which $\beta_{min} \leq \beta_{max}$, the operator $*$ can be either $=$ or one of the two subset operators, strict inclusion ($\subset$) or possibly improper inclusion ($\subseteq$), the term t is either a constant set term or a variable in $\mathbf{X}$ that is bound to a set term in ϕ. Rule (1) of the example in the Introduction is a tuple count constraint while rules (2) and (3) are group count constraints. The formal semantics of count constraints is described in [1].

A relation r satisfies a count constraint C (and we write $r \models C$) if the evaluation of C on r is equal to true. Accordingly, r satisfies a set of count constraints $\mathbf{C}$ (and we write $r \models \mathbf{C}$) if for each $C \in \mathbf{C}$, $r \models C$.

As stated in the proposition below, checking count constraint satisfaction may require exponential time. For space reasons the proof is omitted and can be found in [1] and [16].

Proposition 1. *Given a count constraint C and a relation r on R, deciding $r \models C$ is in* EXP.

Particularly interesting are count constraints for which satisfaction can be tested in polynomial time.

Inverse OLAP Problem. *Given a set of count constraints C on R, the Inverse OLAP problem consists of deciding whether there exists a relation r on $\mathcal{R}$ such that $r \models C$.*

Proposition 2. *The Inverse OLAP problem is* NEXP*-complete.*

The proof is reported in [1] and [16]. We point out that, in our general setting, we are considering the so-called "combined complexity" [17]: both the number of attributes, the size of constraints and the domain sizes are not fixed and are part of the input. The "program complexity" version of the problem consists of considering domain sizes as constants. On the other hand, the "data complexity" version of the inverse OLAP problem fixes the number of attributes and the size of constraints and considers domain sizes as the only problem input. The NEXP-completeness of Inverse OLAP under combined complexity is not surprising: in fact, even the simple evaluation of single clause DATALOG programs is known to be EXP-complete [18]. What is really surprising is that inverse OLAP is NEXP-complete also under data complexity. This result depends mainly on group count constraints, that introduce the typical high complexity of data mining problems.

3 A Motivating Example

We refer to a classical example of point-of-sales transaction star schema. The attributes of U are: T (Transaction), I (Item), B (Brand), S (Store), A (Area) — their (finite) domains can be suitably defined. We are also given the following functional dependencies (FDs): $T \rightarrow S$, $S \rightarrow A$. It turns out that $\{T, I, B\}$ is the relation key. The domains of the attributes are denoted by $\mathcal{D}_T$, $\mathcal{D}_I$ and so on. We next present a number of meaningful count constraints that clarify their usage. To simplify the notation, all low-case letter variables are intended to be universally quantified.

(i): Enforcing FDs and relation key

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

For instance the FD $T \rightarrow S$ can be expressed as follows:

$$\mathcal{D}_{\mathtt{T}}(\mathtt{t}) \ \rightarrow \ 0 \leq \#(\{\mathtt{S} : \exists \mathtt{I}, \mathtt{B}, \mathtt{A}\, \mathcal{R}(\mathtt{t}, \mathtt{I}, \mathtt{B}, \mathtt{S}, \mathtt{A})\}) \leq 1$$

The relation key $\{T, I, B\}$ can be enforced as:

$$\mathcal{D}_{\mathtt{T}}(\mathtt{t}) \wedge \mathcal{D}_{\mathtt{I}}(\mathtt{i}) \wedge \mathcal{D}_{\mathtt{B}}(\mathtt{b}) \rightarrow 0 \leq \#(\{\mathtt{S}, \mathtt{A} : \mathcal{R}(\mathtt{t}, \mathtt{i}, \mathtt{b}, \mathtt{S}, \mathtt{A}))\}) \leq 1$$

(ii): Enforcing the overall number of tuples

There must be between 50000 and 100000 tuples in any instance of R:

$$true \rightarrow \ 50000 \leq \#(\{\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{S}, \mathtt{A} : \mathcal{R}(\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{S}, \mathtt{A})\}) \leq 100000$$

(iii): Enforcing the total number of transactions in an area

There must be between 1000 and 2000 transactions in every region, except in "Cal" for which the upper bound is increased to 9000:

$$true \ \rightarrow 1000 \leq \#(\{\mathtt{T} : \exists \mathtt{I}, \mathtt{B}, \mathtt{S}\, \mathcal{R}(\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{S}, \text{"}\mathtt{Cal}\text{"})\}) \leq 9000;$$

$$\mathcal{D}_{\mathtt{A}}(\mathtt{a}) \wedge \mathtt{a} \neq \text{"}\mathtt{Cal}\text{"} \ \rightarrow 1000 \leq \#(\{\mathtt{T} : \exists \mathtt{I}, \mathtt{B}, \mathtt{S}\, \mathcal{R}(\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{S}, \mathtt{a})\}) \leq 2000.$$

If we wish to enforce the above transaction constraint in every store of an area, we can use the dimension hierarchy domain $\mathcal{D}_{S,A}$:

$$\mathcal{D}_{\mathtt{S},\mathtt{A}}(\mathtt{s}, \text{"}\mathtt{Cal}\text{"}) \rightarrow 1000 \leq \#(\{\mathtt{T} : \exists \mathtt{I}, \mathtt{B}\, \mathcal{R}(\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{s}, \text{"}\mathtt{Cal}\text{"})\}) \leq 9000;$$

$$\mathcal{D}_{\mathtt{S},\mathtt{A}}(\mathtt{s}, \mathtt{a}) \wedge \mathtt{a} \neq \text{"}\mathtt{Cal}\text{"} \ \rightarrow 1000 \leq \#(\{\mathtt{T} : \exists \mathtt{I}, \mathtt{B}\, \mathcal{R}(\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{s}, \mathtt{a})\}) \leq 2000\ .$$

(iv): 1-arity group count constraints

Both the set of items $i = \{[a], [b], [c]\}$ and $j = \{[b], [c], [d]\}$) must be present in at least 100 and in at most 200 transactions, whereas every other set s of items cannot appear in more than 15 transactions if s contains more than 10 elements or 20 otherwise, except for all subsets of i and j that have no limits (for space reasons we write below i and j to represent the two constant set terms):

$$\mathtt{x} = i \vee \mathtt{x} = j \rightarrow 100 \leq \#(\{\mathtt{T} : \mathtt{x} \subseteq \{\mathtt{I} : \exists \mathtt{B}, \mathtt{S}, \mathtt{A}\, \mathcal{R}(\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{S}, \mathtt{A})\}\}) \leq 200;$$

$$\mathtt{x} \subseteq \{\mathtt{I} : \mathcal{D}_{\mathtt{I}}(\mathtt{I})\} \wedge \mathtt{x} \not\subset i \wedge \mathtt{x} \not\subset j \wedge \#(\mathtt{x}) \leq 10 \ \rightarrow$$
$$0 \leq \#(\{\mathtt{T} : \mathtt{x} \subseteq \{\mathtt{I} : \exists \mathtt{B}, \mathtt{S}, \mathtt{A}\, \mathcal{R}(\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{S}, \mathtt{A})\}\}) \leq 20;$$

$$\mathtt{x} \subseteq \{\mathtt{I} : \mathcal{D}_{\mathtt{I}}(\mathtt{I})\} \wedge \mathtt{x} \not\subset i \wedge \mathtt{x} \not\subset j \wedge \#(\mathtt{x}) > 10 \ \rightarrow$$
$$0 \leq \#(\{\mathtt{T} : \mathtt{x} \subseteq \{\mathtt{I} : \exists \mathtt{B}, \mathtt{S}, \mathtt{A}\, \mathcal{R}(\mathtt{T}, \mathtt{I}, \mathtt{B}, \mathtt{S}, \mathtt{A})\}\}) \leq 15.$$

Note that the first of the above constraints has a disjunction in the left hand side: it is only a shorthand to represent two constraints having the same right hand side.

The above constraints define an instance of an IFM problem [13–15] for which, in addition to fixing support constraints for a number of pre-defined itemsets (typically the frequent ones, in this case i and j), there are generic support constraints for all other itemsets (the unfrequent ones). The example confirms that Inverse OLAP is a powerful extension of IFM.

(v): 2-arity group count constraints

There must be at least 100 and at most 200 transactions containing an item "sm" (smartphone) of the brand "nd" (ndrangtung), whereas the same set of pairs of item and brand are sold together in at most 10 transactions, except for the ones containing the

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

pair ("sm","nd") for which the limit is 50 (for space reasons we write t to represent the singleton constant set term $\{$["sm","nd"]$\}$):

$$true \rightarrow 100 \leq \#(\{\mathtt{T} : t \subseteq \{\mathtt{I},\mathtt{B} : \exists \mathtt{S},\mathtt{A}\, \mathcal{R}(\mathtt{T},\mathtt{I},\mathtt{B},\mathtt{S},\mathtt{A}))\}\}) \leq 200;$$

$$\mathtt{x} \subseteq \{\mathtt{I},\mathtt{B} : \mathcal{D}_{\mathtt{I}}(\mathtt{I}) \wedge \mathcal{D}_{\mathtt{B}}(\mathtt{B})\} \wedge t \subset \mathtt{x} \rightarrow$$
$$0 \leq \#(\{\mathtt{T} : \mathtt{x} \subseteq \{\mathtt{I},\mathtt{B} : \exists \mathtt{S},\mathtt{A}\, \mathcal{R}(\mathtt{T},\mathtt{I},\mathtt{B},\mathtt{S},\mathtt{A}))\}\}) \leq 50;$$

$$\mathtt{x} \subseteq \{\mathtt{I},\mathtt{B} : \mathcal{D}_{\mathtt{I}}(\mathtt{I}) \wedge \mathcal{D}_{\mathtt{B}}(\mathtt{B})\} \wedge t \not\subset \mathtt{x} \rightarrow$$
$$0 \leq \#(\{\mathtt{T} : \mathtt{x} \subseteq \{\mathtt{I},\mathtt{B} : \exists \mathtt{S},\mathtt{A}\, \mathcal{R}(\mathtt{T},\mathtt{I},\mathtt{B},\mathtt{S},\mathtt{A}))\}\}) \leq 10.$$

(Recall that $\subset$ denotes strict subset relationship.) The above constraints define an instance of an Inverse Frequent Itemset problem in which classical itemsets are replaced by sets of object pairs.

4 A Step towards Aggregate Data Exchange

Data exchange [5–7] is the problem of migrating a data instance from a source schema to a target schema such that the materialized data on the target schema satisfies the integrity constraints specified by it. The classical data exchange setting is: $(S, T, \Sigma_{st}, \Sigma_t)$, where S is the source relational database schema, T is the target schema, Σ_t are dependencies on the target scheme T and Σ_{st} are source-to-target dependencies.

The dependencies in Σ_{st} mapping data from the source to the target schema are TGDs (Tuple Generating Dependencies) and have the following format: $\forall \mathbf{X}(\, \phi_S(\mathbf{X}) \rightarrow \exists \mathbf{Y}\ \ \psi_T(\mathbf{X}, \mathbf{Y})\,)$, where $\phi_S(\mathbf{X})$ and $\psi_T(\mathbf{X}, \mathbf{Y})$ are formula on S and T, respectively, and $\mathbf{X}$, $\mathbf{Y}$ are lists of variables.

Dependencies in Σ_t specify constraints on the target schema, which the imported data must satisfy, and can be either TGDs or EGDs (Equality Generating Dependencies) having the form $\forall \mathbf{X}(\, \psi_T(\mathbf{X}) \rightarrow x_1 = x_2\,)$, where x_1 and x_2 are variables in $\mathbf{X}$.

It is easy to see that a generic EGD can be formulated by the following count constraint: $\forall \mathbf{X}'\ (\, true \rightarrow 0 \leq \#(\{y : \psi(y, \mathbf{X}')\}\,) \leq 1)$, where $\mathbf{X}'$ contains all variables in $\mathbf{X}$ except x_1 and x_2; moreover, y replaces both x_1 and x_2 in ψ.

Aggregate data exchange for preserving privacy

The target relational database scheme consists of a unique relation scheme, that is the one used in Section 3: $\mathcal{R}(T, I, B, S, A)$. Recall that the meaning of the attributes is: T (Transaction), I (Item), B (Brand), S (Store), A (Area), and that the following FDs hold: $T \rightarrow S$, $S \rightarrow A$.

The source relational database scheme consists of three relation schemes: $\mathcal{TR}(T, I, B)$, $\mathcal{ST}(T, S)$ and $\mathcal{AR}(S, A)$. Observe that this scheme is the normalized version of the target scheme.

We want the target relation to be the natural join of the source relation but, for privacy reasons, the associations between transactions and pairs of item and brand must be perturbed: the transactions IDs of the same store are permuted. For instance, if the store s has n transactions $t_1, \ldots, t_n$, the block of item-brand pairs of a transaction t_i are moved to a transaction t_j, then the block of t_j is moved to another transaction and so on.

Let us first use the classical setting to implement two natural joins of the source relations instead of only one so that we can later perform a permutation of transactions inside every store:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

$$\mathcal{TR}(\mathtt{t},\mathtt{i},\mathtt{b}) \wedge \mathcal{ST}(\mathtt{t},\mathtt{s}) \wedge \mathcal{AR}(\mathtt{s},\mathtt{a}) \rightarrow \exists \mathtt{T}\, \mathcal{R}(\mathtt{T},\mathtt{i},\mathtt{b},\mathtt{s},\mathtt{a});$$
$$\mathcal{ST}(\mathtt{t},\mathtt{s}) \rightarrow \exists \mathtt{I},\mathtt{B},\mathtt{A}\, \mathcal{R}(\mathtt{t},\mathtt{I},\mathtt{B},\mathtt{s},\mathtt{A}).$$

We now use a count constraint to enforce that the total number of tuples in $\mathcal{TR}$ is equal to the total number of tuples in $\mathcal{R}$ so that the target relation cannot store additional tuples:

$$\mathtt{x} = \#(\{\mathtt{T},\mathtt{I},\mathtt{B} : \mathcal{TR}(\mathtt{T},\mathtt{I},\mathtt{B})\}) \rightarrow \mathtt{x} = \#(\{\mathtt{T},\mathtt{I},\mathtt{B},\mathtt{S},\mathtt{A} : \mathcal{R}(\mathtt{T},\mathtt{I},\mathtt{B},\mathtt{S},\mathtt{A})\}).$$

So we have lost the correspondence between transactions in the source and in the target scheme; but the following constraint imposes that the original structure of transactions is preserved modulo permutation of transactions IDs:

$$\mathcal{ST}(\mathtt{t},\mathtt{s}) \wedge \mathtt{x} = \{\mathtt{I},\mathtt{B} : \mathcal{TR}(\mathtt{t},\mathtt{I},\mathtt{B})\}) \rightarrow \exists \mathtt{T}\, (\, \mathtt{x} = \{\mathtt{I},\mathtt{B} : \exists \mathtt{A}\, \mathcal{R}(\mathtt{T},\mathtt{I},\mathtt{B},\mathtt{s},\mathtt{A})\}\,).$$

Data exchange to an OLAP scheme

We now assume that the relation $\mathcal{R}(T, I, B, S, A)$ represents the source scheme. The target scheme is an OLAP scheme $\mathcal{SN}(S, I, B, N)$ that, for every store, represents in N the total number of item-brand pairs that are in all transactions of that store.

We aggregate data in the target relation using a count predicate in the following constraint:

$$\mathtt{n} = \#(\{\mathtt{T} : \mathcal{R}(\mathtt{T},\mathtt{i},\mathtt{b},\mathtt{s},\mathtt{a})\}) \rightarrow \mathcal{SN}(\mathtt{s},\mathtt{i},\mathtt{b},\mathtt{n}).$$

A final count constraint imposes that the target relation cannot store additional tuples:

$$\mathtt{x} = \#(\{\mathtt{S},\mathtt{I},\mathtt{B} : \exists \mathtt{T},\mathtt{A}\, \mathcal{R}(\mathtt{T},\mathtt{I},\mathtt{B},\mathtt{S},\mathtt{A})\}) \rightarrow \mathtt{x} = \#(\{\mathtt{S},\mathtt{I},\mathtt{B} : \exists \mathtt{N}\, \mathcal{SN}(\mathtt{S},\mathtt{I},\mathtt{B},\mathtt{N})\}\,).$$

5 Conclusion

We have introduced a new type of constraints, called count constraints, and a new inverse mining problem, called Inverse OLAP, that is a powerful extension of Inverse Frequent itemsets Mining: given a star schema consisting of a unique fact table and a number of count constraints, does there exist a satisfying relation? The new problem turns out to be NEXP complete under various conditions: combined complexity, program complexity and data complexity. We have also shown that count constraints can be used for performing aggregate data exchange.

We conclude by mentioning that, despite the high complexity of the Inverse OLAP problem, an approximate solution can be found in a limited amount of time in some practical situations even for large instances, by adopting and extending classical techniques used for solving large-scale linear programming. In [19] one of such techniques, called column-generation linear programming, is applied to solve an IFM problem (indeed non just the decision problem but the actual construction of a satisfying transaction database), that can be though of as a special case of inverse OLAP. Such techniques are capable of handling instances with several hundreds of items. In addition our current research is devoted to single out cases for which complexity of Inverse OLAP becomes polynomial under the data complexity by considering particular cases of count constraints.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

References

1. Saccà, D., Serra, E., Guzzo, A.: Count constraints and the inverse olap problem: Definition, complexity and a step toward aggregate data exchange. In: FoIKS. (2012) 352–369
2. Zhang, X., Ozsoyoglu, Z.M.: Implication and referential constraints: A new formal reasoning. IEEE Trans. on Knowledge and Data Engineering **9** (1997) 894–910
3. Rosati, R.: On the decidability and finite controllability of query processing in databases with incomplete information. In: PODS. (2006) 356–365
4. Cosmadakis, S.S., Kanellakis, P.C., Vardi, M.Y.: Polynomial-time implication problems for unary inclusion dependencies. J. of the ACM **37** (1990) 15–46
5. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: Semantics and query answering. In: In ICDT. (2003) 207–224
6. Arenas, M., Barcel, P., Fagin, R., Libkin, L.: Locally consistent transformations and query answering in data exchange. In: PODS'04. (2004) 229–240
7. Fagin, R., Kolaitis, P.G., Popa, L.: Data exchange: getting to the core. ACM Trans. Database Syst. **30**(1) (2005) 174–210
8. Chaudhuri, S., Dayal, U.: An overview of data warehousing and OLAP technology. SIGMOD Record **26**(1) (1997) 65–74
9. Golfarelli, M., Rizzi, S.: Data Warehouse Design: Modern Principles and Methodologies. Mac Graw Hill (2009)
10. Arasu, A., Kaushik, R., Li, J.: Data generation using declarative constraints. In: Proceedings of the 2011 international conference on Management of data. SIGMOD '11, New York, NY, USA, ACM (2011) 685–696
11. Faber, W., Pfeifer, G., Leone, N., Dell'Armi, T., Ielpa, G.: Design and implementation of aggregate functions in the dlv system. TPLP **8**(5-6) (2008) 545–580
12. Syrjänen, T.: Logic Programs and Cardinality Constraints: Theory and Practice. Doctoral dissertation, TKK Dissertations in Information and Computer Science TKK-ICS-D12, Helsinki University of Technology, Department of Information and Computer Science (2009)
13. Mielikainen, T.: On inverse frequent set mining. In Society, I.C., ed.: Proc. of 2nd Workshop on Privacy Preserving Data Mining (PPDM). (2003) 18–23
14. Calders, T.: Computational complexity of itemset frequency satisfiability. In: PODS. (2004) 143–154
15. Calders, T.: The complexity of satisfying constraints on databases of transactions. Acta Inf. **44**(7-8) (2007) 591–624
16. Saccà, D., Serra, E., Guzzo, A.: Appendix to [1]. In: http://sacca.deis.unical.it/#view=object&format=object&id=960/gid=160. (2012)
17. Vardi, M.Y.: The complexity of relational query languages (extended abstract). In: STOC. (1982) 137–146
18. Gottlob, G., Papadimitriou, C.H.: On the complexity of single-rule datalog queries. In: LPAR. (1999) 201–222
19. Guzzo, A., Moccia, L., Saccà, D., Serra, E.: Solving inverse frequent itemset mining with infrequency constraint via large-scale linear programs. In: http://sacca.deis.unical.it/#view=object&format=object&id=981/gid=160. (2011)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Individual Mobility Profiles: Methods and Application on Vehicle Sharing*

Roberto Trasarti[1], Fabio Pinelli[2], Mirco Nanni[1], and Fosca Giannotti[1]

[1] KDDLab, ISTI-CNR, Pisa, Italy: email name.surname@isti.cnr.it
[2] IBM Research Lab, Dublin, Ireland: fabiopin@ie.ibm.com

Abstract. In this paper we present a methodology for extracting mobility profiles of individuals from raw digital traces (in particular, GPS traces), and study criteria to match individuals based on profiles. We instantiate the profile matching problem to a specific application context, namely proactive car pooling services, and therefore develop a matching criterion that satisfies various basic constraints obtained from the background knowledge of the application domain. In order to evaluate the impact and robustness of the methods introduced we present an experiment which is performed on a massive dataset containing GPS traces of private cars.

1 Introduction

The traditional use of mobility data, for instance in the context of urban traffic monitoring and transportation planning, mainly focuses on inferring simple measurements and aggregations, such as density of traffic, and car flows on road segments. Despite the great attention that this area has attracted, current work on mobility analysis largely neglects a key element that lies in between single trajectories and a whole population, i.e. the individual person, with his/her regularities and habits, that can be differed from the population. In fact, analysing individuals (rather than just large groups) provides the basis for an understanding of systematic mobility, as opposed to occasional movements, which is fundamental in some mobility planning applications, e.g. public transport. The standard approach adapts classical distance-based algorithms and defines ad hoc distances for trajectory data [7], possibly with limited ad hoc refinements [3] or ad hoc solutions include variants of model-based clustering [4], collective movements detection methods [6], and others. As opposed to existing solutions, in our proposal, already published in [1], the evaluation of similarity between individuals is not realized as a direct comparison of trajectories. Instead,we propose a two-phase process: first an individual-centered mobility model extraction; then a population-wide analysis based on the individual models. Our framework can be seen as a new approach in the learning paradigm since it provides a local-to-global analysis.

2 Mobility profiles extraction

The daily mobility of each user can be essentially summarized by a set of single trips that the user performs during the day. When trying to extract a *mobility profile* of users,

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

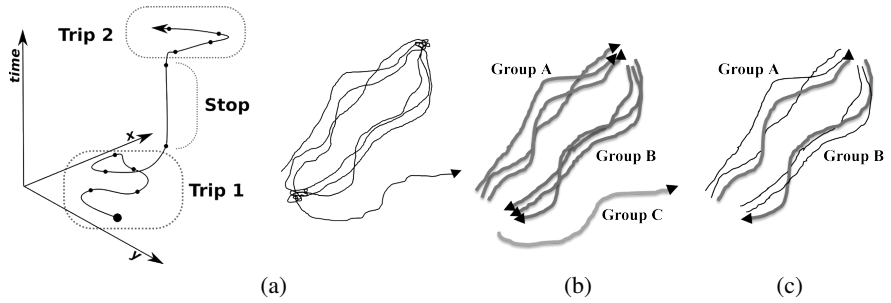


Fig. 1. Mobility profile extraction process: (a) trip identification; (b) group detection/outlier removal; (c) selection of representative mobility profiles.

our interest is in the trips that are part of their habits, therefore neglecting occasional variations that divert from their typical behavior. Therefore in order to identify the individual mobility profiles of users from their GPS traces, the following steps will be performed - see Figure 1:

1. divide the whole history of the user into trips (Figure 1(a))
2. group trips that are similar, discarding the outliers (Figure 1(b))
3. from each group, extract a set of representative trips, to be used as mobility profiles (Figure 1(c)).

2.1 Mobility profile definitions

Trips. The history of a user is represented by the set of points in space and time recorded by their mobility device:

Definition 1 (User history). *The user history is defined as an ordered sequence of spatio-temporal points* $H = \langle p_1 \dots p_n \rangle$ *where* $p_i = (x, y, t)$ *and* x, y *are spatial coordinates and* t *is an absolute timepoint.*

This continuous stream of information contains different trips made by the user, therefore in order to distinguish between them we need to detect when a user stops for a while in a place. This point in the stream will correspond to the end of a trip and the beginning of the next one. In this paper we adopt the latter for computational efficiency reasons. Thus we look for points that change only in time; i.e. they keep the same spatial position for a certain amount of time quantified by the temporal threshold $th^{stop}_{temporal}$. Specularly, a spatial threshold $th^{stop}_{spatial}$ is used to remove both the noise introduced by the imprecision of the device and the small movements that are of no interest for a particular analysis.

Definition 2 (Potential stops). *Given the history* H *of a user and the thresholds* $th^{stop}_{spatial}$ *and* $th^{stop}_{temporal}$*, a* potential stop *is defined as a maximal subsequence* S *of the user's history* H *where the points remain within a spatial area for a certain period of time:* $S = \langle p_m \dots p_k \rangle \,|0 < m \leq k \leq n \wedge \forall_{m \leq i \leq k} Dist(p_m, p_i) \leq th^{stop}_{spatial} \wedge Dur(p_m, p_k) \geq th^{stop}_{temporal}$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

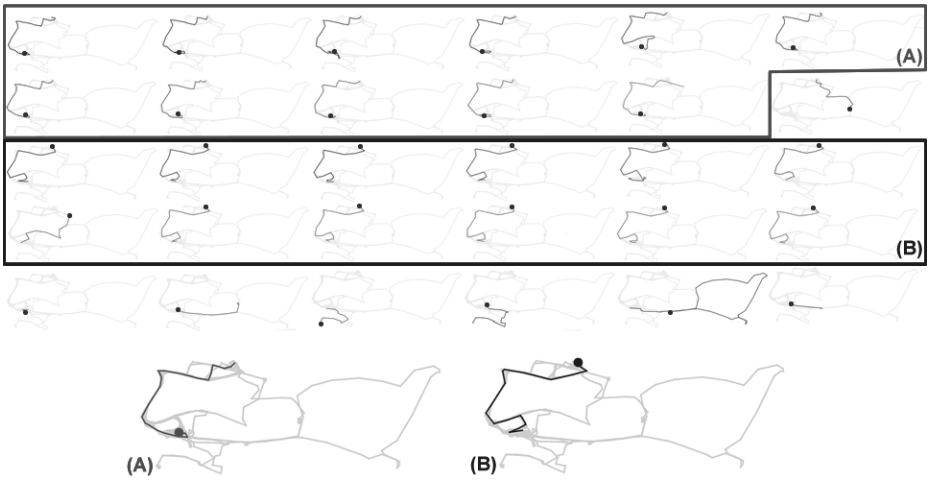


Fig. 2. Trajectories of a user and the corresponding groups and routines extracted (A and B). Of the 30 trips, 11 are part of group A, and 12 of group B, while the remaining 7 are noise. The two routines are spatially similar, yet move in opposite directions (points represent the end of trips), i.e., south (A) vs. north (B).

where $Dist$ is the Euclidean distance function defined between the spatial coordinates of the points, and Dur is the difference in the temporal coordinates of the points. Potential stops can overlap with each other (yet, none of them can completely contain the other, for the maximality condition), making it difficult to use them as a basis for further analysis. In order to avoid this, a criterion of *early selection* is adopted to remove any overlaps:

Definition 3 (Actual stops). *Given a sequence of potential stops $S_{set} = \langle S_1, \ldots, S_N \rangle$, sorted by starting time (i.e., $S \leq S' \Leftrightarrow S = \langle (x, y, t), \ldots \rangle \wedge S' = \langle (x', y', t'), \ldots \rangle \wedge t \leq t'$), the corresponding sequence of* actual stops *$ActS$ is defined as the minimal sequence of potential stops such that:*

1. $S_1 \in ActS$
2. *if* $S_i \in ActS \wedge k = \min\{j | j > i \wedge S_j \cap S_i = \emptyset\} < \infty \quad \Rightarrow \quad S_k \in ActS$

We indicate with $\overline{S} = \langle S_1 \ldots S_t \rangle$ the set of all actual stops over H. Once we have found the stops in the users history we can identify the trips:

Definition 4 (Trip). *A trip is defined as a subsequence T of the user's history H between two consecutive actual stops in the ordered set $\overline{S}$ or between an actual stop and the first/last point of H (i.e., p_1 or p_n):*

- $T = \langle p_m, \ldots, p_k \rangle \, | 0 < m \leq k \leq n \wedge \exists i (S_i = \langle \ldots, p_m \rangle \wedge S_{i+1} = \langle p_k, \ldots \rangle)$, *or*
- $T = \langle p_1, \ldots, p_m \rangle \, | 0 < m \leq n \wedge \exists i (S_i = \langle p_m, \ldots \rangle)$, *or*
- $T = \langle p_k, \ldots, p_n \rangle \, | 0 < k \leq n \wedge \exists i (S_i = \langle \ldots, p_k \rangle)$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

The set of extracted trips $\overline{T} = \langle T_1 \dots T_c \rangle$ in Fig. 1(a), are the basic steps to create the user mobility profile. Notice that the thresholds $th^{stop}_{spatial}$ and $th^{stop}_{temporal}$ are the knobs for expressing specific analytical requirements.

Trip groups. Our objective is to use the set of trips of an individual user to find his/her routine behaviors. We do this by grouping together similar trips based on concepts of spatial distance and temporal alignment, with corresponding thresholds for both the spatial and temporal components of the trips. In order to be defined as *routine*, a behavior needs to be supported by a significant number of similar trips. The above ideas are formalized as follows:

Definition 5 (Trip Group). *Given a set of trips $\overline{T}$, spatial and temporal thresholds $th^{group}_{spatial}$ and $th^{group}_{temporal}$, a spatial distance function $\delta : \overline{T}^2 \to \mathcal{R}$, a temporal alignment constraint $\alpha : \overline{T}^2 \times \mathcal{R} \to \mathcal{B}$ between pairs of trips, and a minimum support threshold $th^{group}_{support}$, a* trip group *for $\overline{T}$ is defined as a subset of trips $g \subseteq \overline{T}$ such that:*

1. $\forall t_1, t_2 \in g . \delta(t_1, t_2) \leq th^{group}_{spatial} \wedge \alpha(t_1, t_2, th^{group}_{temporal})$;
2. $|g| \geq th^{group}_{support}$.

Condition 1 requires that the trips in a group are approximately co-located, both in space and time, while condition 2 requires that the group is sufficiently large. Again, the thresholds are the knobs that the analyst will progressively tune the extraction process with.

Mobility Profile. Each group obtained in the previous step represents the typical mobility habit of a user, i.e., one of his/her routine movements. Here we summarize the whole group by choosing the central element of such a group:

Definition 6 (Routine). *Given a trip group g and the distance function δ used to compute it, its* routine *is defined as the medoid of the set, i.e.:*

$$routine(g, \delta) = \arg\min_{t \in g} \sum_{t' \in g \setminus \{t\}} \delta(t, t')$$

Notice that the temporal alignment is always satisfied over each pair of trips in a group, therefore the alignment relation α does not appear in the definition. Now we are ready to define the users mobility profile.

Definition 7 (Mobility Profile). *Given a set of trip groups G of a user and the distance function δ used to compute them, the user's* mobility profile *is defined as his/her corresponding set of routines:*

$$profile(G, \delta) = \{routine(g, \delta) \mid g \in G\}$$

Mobility profile construction. The definitions provided in the previous section were kept generic w.r.t. the distance function δ. Different choices can satisfy different needs, possibly both conceptually (which criteria define a good group/routine assignment) and pragmatically (for instance, simpler criteria might be preferred for the sake of

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

scalability). Obviously, the results obtained by different instantiations can vary greatly. Our proposal is to use a clustering method to carry out this task. We choose the clustering algorithm for trajectories proposed in [3], consisting of two steps. First, a density-based clustering is performed, thus removing noisy elements and producing dense – yet, possibly extensive – clusters. Secondly, each cluster is split through a bisection k-medoid procedure. Such method splits the dataset into two parts through k-medoid (a variant of k-means) with $k = 2$, then the same splitting process is recursively applied to each sub-group. Recursion stops when each resulting sub-cluster is compact enough to fit within a distance threshold of its medoid, by removing sub-clusters that are too small. The bisection k-medoid procedure guarantees that requirements 1 and 2 of Definition 5 are satisfied. The clustering method adopted is parametric w.r.t. a repertoire of similarity functions, that includes: *Ends* and *Starts* functions, comparing trajectories by considering only their last (respecively, first) points; *Route similarity*, comparing the paths followed by trajectories from a purely spatial viewpoint (time is not considered); *Synchronized route similarity*, similar to Route similarity but considering also time.

2.2 Profiling GPS-equipped vehicles

In this section we present the results of our method applied to a real dataset of GPS observations of 2,107 real car users in Tuscany in a time period of 12 days covering different kind of territories such as urban and suburban areas. This is a sample of data obtained by a private company employed specifically as a service for insurance companies and other clients called *octotelematics* [2]. The process is implemented using the data mining query language provided by the M-Atlas system [8]. We processed this dataset of observations using the *mobility profile construction* algorithm, with the following parameters:

δ **and** α : we adopted the *route similarity* function described in [3] as spatial distance function (δ). The route similarity function performs an alignment between points of the trajectories (trips) that are going to be compared, and then computes the sum of distances between corresponding points. In addition, we adopted a temporal alignment constraint (α) which simply computes the temporal distance between the starting points of the two trips, and compares it against the temporal threshold.

$th^{stop}_{spatial}$ **and** $th^{stop}_{temporal}$: 50 meters and 1 hour, this means that we consider a stop when a user stays with his/her car in an area of 50 m^2 for at least one hour. Single trips of a user are thus the movements between these stops.

$th^{group}_{spatial}$ **and** $th^{group}_{temporal}$: 250 meters and 1 hour, we want to group trips which are *similar* considering a maximum of 250 meters and a temporal alignment of 1 hour.

$th^{group}_{support}$: 4 trips, only the groups with at least 4 trips survive the pruning process, the others are not considered interesting enough for the mobility profiles.

An example of how the *mobility profile construction* works is shown in Fig.2. As can be seen, two main routes are frequently repeated, each time with small variations. In addition, they appear to represent symmetric trips, such as home-to-work and work-to-home routine movements. The corresponding mobility profiles are depicted at the bottom of the figure. Notice that seven user trips were occasional trips that did not fit any consistent habit, and therefore were (correctly) filtered out by our algorithm.

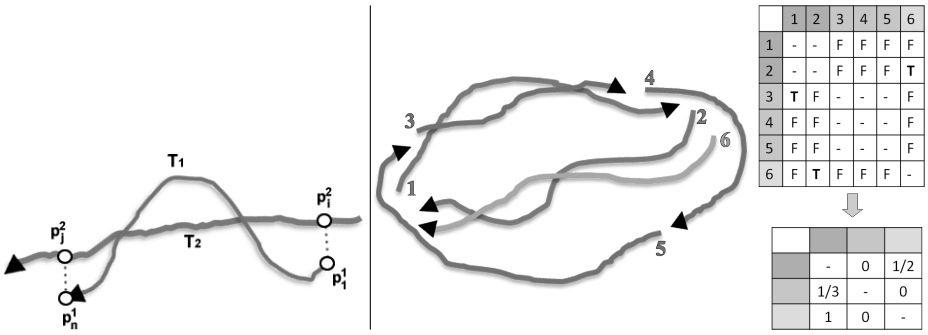
Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Globally during the execution of the algorithm, a set of 46,163 trips is generated and the result of the mobility profile construction is a set of 1,504 routines that form 919 mobility profiles (i.e., for 43.6% of the 2,107 users a profile was extracted). For space reason we don't report the complete analysis presented in [1] on how the method is effected by the parameters and how much the profiles extracted remain persistent and stable in time.

3 Mobility Profile matching

In this paper we focus on a *car pooling* application aimed at identifying pairs of users that could most likely share their vehicle for one or more of their routine trips. The service might be deployed as a system that provides pro-active suggestions to facilitate the matching process, without the need for the user to explicitly describe (and update) the trips of interest. The starting point of this analysis is the set of representative trips which make up the user mobility profiles. These mobility profiles represent their different typical behaviors, and by comparing them, we can understand if a user can be *served* by another user.

Definition 8 (Routine containment). *Given two mobility routines* $T_1 = \langle p_1^1 \dots p_n^1 \rangle$ *and* $T_2 = \langle p_1^2 \dots p_m^2 \rangle$*, and thresholds* $th_{distance}^{walking}$ *and* $th_{time}^{wasting}$*, we say that* T_1 is contained in T_2*, denoted* $contained(T_1, T_2, th_{distance}^{walking}, th_{time}^{wasting})$
$iff : contained(T_1, T_2, th_{distance}^{walking}, th_{time}^{wasting}) \equiv \exists i, j \in \mathcal{N} \; | 0 < i \leq j \leq m \; \wedge$
$Dist(p_1^1, p_i^2) + Dist(p_n^1, p_j^2) \leq th_{distance}^{walking} \wedge Dur(p_1^1, p_i^2) + Dur(p_n^1, p_j^2) \leq th_{time}^{wasting}$



	1	2	3	4	5	6
1	-	-	F	F	F	F
2	-	-	F	F	F	**T**
3	**T**	F	-	-	-	F
4	F	F	-	-	-	F
5	F	F	-	-	-	F
6	F	**T**	F	F	F	-

	-	0	1/2
	1/3	-	0
	1	0	-

Fig. 3. On the left: example of routine containment test for Definition 8. On the right: there is an example of the mobility profile matching process where the routines of the same color belong to the same mobility profile: the routines matrix-containment (top) and the profile share-ability matrix (bottom).

Thresholds $th_{distance}^{walking}$ and $th_{time}^{wasting}$ represent the total spatial and temporal distances allowed between the two routines in space and time, in other words:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

$th^{walking}_{distance}$: represents the maximum distance the user which *is served* could walk to reach the meeting point and then to reach their final destination at the end of the trip.

$th^{wasting}_{time}$: represents the maximum delay the user which *is served* allows, considering the departure and the arrival time.

It is important to note that the $contains$ relation is not reflexive because one trip can include the other but not vice versa. This is a basic requirement in the car pooling application because the destinations of the user which *serves* the other can be very far from the destination of the one who is *served* (Fig.3(left)). Extending the definition to the mobility profiles of the users, we can compute the share-ability level of each pair of users:

Definition 9 (Mobility profile share-ability). *Given two mobility profiles $\tilde{T}_1$ and $\tilde{T}_2$, and thresholds $th^{walking}_{distance}$ and $th^{wasting}_{time}$, the* Mobility profile share-ability *measure between $\tilde{T}_1$ and $\tilde{T}_2$ is defined as the fraction of routines in $\tilde{T}_1$ which are contained in at least one routine in $\tilde{T}_2$:*

$$profileShare(\tilde{T}_1, \tilde{T}_2, th^{walking}_{distance}, th^{wasting}_{time}) = \frac{\left| \left\{ p \in \tilde{T}_1 \mid \exists q \in \tilde{T}_2, contained(p,q,th^{walking}_{distance}, th^{wasting}_{time}) \right\} \right|}{\left| \tilde{T}_1 \right|}$$

By applying this definition to all possible pairs of users (i.e., to their corresponding profiles) we can build a matrix of share-ability, thus expressing how good the match of each pair is. The algorithm first builds a *routine containment matrix* over single mobility routines, (ii) then the results corresponding to each pair of users are collapsed to form a mobility profile share-ability matrix, by applying the Definition 9. A visual example of the result is shown in Fig.3(right).

3.1 The car pooling service with GPS data

We used Algorithm presented in previous section to perform the matching process on our data with different parameter settings. The results in Figure 4(right) show how the performances are affected, in terms of percentage routines and mobility profiles that have at least one match. Note that by allowing a *walking distance* of 5 km and a *wasting time* of 1 hour, 89% of profiled users have (at least) one match, which decreases to 66% if the *wasting time* becomes half an hour. Figure 4(left) shows two examples of matching between two users. The red user can be served by the violet user on the basis of the routines shown. In the two examples it is interesting to see that in the first case (A), the starts and ends of the routines are quite close, therefore these users can both serve or be served by each other; in the second case (B) the relation is unidirectional, since the red routine ends much earlier than the other, and therefore the *contain* relation does not hold in the opposite direction.

Considering a hypothetical car pooling service built on top of the proposed method, using a *walking distance* of 2.5 km and a *wasting time* of 1 hour, we can calculate some statistics regarding the potential impact of the service. In fact 684 users, corresponding

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

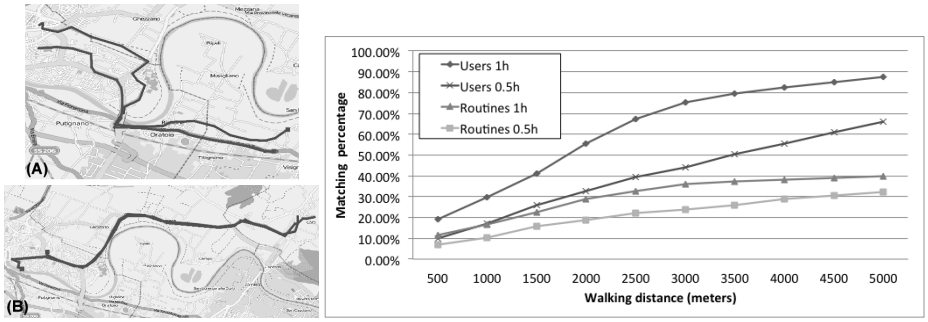


Fig. 4. Examples of routine containment: red routines are contained in the violet ones (left). Matching percentages of users (upper curves) and routines (lower curves) for different settings of the spatial and temporal thresholds (right).

to 32.4% of participants, receive at least one indication of a possible host for one of their routines. This means that if everybody takes the opportunity of sharing a / their car using this system, traffic could be decreased significantly. As previously mentioned, one advantage of the system is that users do not need to manually declare their common trips (indeed, routines are automatically detected), which is a major flaw of current car pooling systems, and probably contributes substantially to their failure. As shown in section 2.2, the system can keep reasonably up-to-date routines and profiles by executing the profiling process once every two weeks (or more), using a temporal sliding window on the data.

References

1. R. Trasarti, F. Pinelli, M. Nanni and F. Giannotti *Mining mobility user profiles for car pooling*. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD 2011.
2. *Octotelematics*. http://www.octotelematics.com/.
3. G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. *Interactive Visual Clustering of Large Collections of Trajectories*. VAST: Symposium on Visual Analytics Science and Technology, 2009.
4. S. Gaffney and P. Smyth. Trajectory clustering with mixture of regression models. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 63–72. ACM, 1999.
5. F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *KDD*, pages 330–339, 2007.
6. P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatio-temporal data. In *Proceedings of 9th International Symposium on Spatial and Temporal Databases (SSTD'05)*, pages 364–381. Springer, 2005.
7. N. Pelekis, I. Kopanakis, I. Ntoutsi, G. Marketos, and Y. Theodoridis. Mining trajectory databases via a suite of distance operators. In *ICDE Workshops*, pages 575–584, 2007.
8. F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, and R. Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *Very Large Database*, 20(5), 2011.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Getting the Best from Uncertain Data: the Correlated Case*

Ilaria Bartolini, Paolo Ciaccia, and Marco Patella

DEIS - Università di Bologna, Italy
{i.bartolini,paolo.ciaccia,marco.patella}@unibo.it

Abstract. In this extended abstract we apply the notion of skyline to the case of probabilistic relations including correlation among tuples. In particular, we consider the relevant case of the x-relation model, consisting of a set of generation rules specifying the mutual exclusion of tuples. We show how our definitions apply to different ranking semantics and analyze the time complexity for the resolution of skyline queries.

1 Introduction

The management of uncertain data has become a very active area of research, due to the huge number of relevant applications it has. Among them, data extraction from the Web, data integration, sensor networks, etc. Consequently, several works have focused on the problem of extending different query types to such scenarios [6, 4]. In [2] we have shown how *skyline queries* can be defined for probabilistic relations, where each tuple has also a probability (confidence) to appear [6, 7]. Although our definition applies to any model of tuple correlation, only the case of independent tuples was analyzed in [2]. In this paper we extend the applicability of skyline queries to the relevant case of *x-relations* [1], where mutual exclusion of tuples is possible. We detail the analysis for the most common ranking semantics (since our skyline is parametric with respect to them) and analyze the time complexity for the resolution of skyline queries.

We adopt the well-known *possible worlds semantics* [4], in which a probabilistic relation R^p represents a set of standard relations, each termed a *possible world*. Compactly, R^p is represented as a triple, $R^p = (R, p, \mathcal{G})$, where R is a relation in the standard sense, i.e., a set of tuples, also called a *deterministic* relation; p is a function that assigns to each tuple $u \in R$ a probability, $p(u) \in (0, 1]$; and $\mathcal{G}$ is a set of *generation rules* that capture correlations among tuples. In the case of the *x-relation* model [1] $\mathcal{G}$ only includes mutual exclusion rules that form a partition of R. In this scenario each $G \in \mathcal{G}$ is also called a *group*, and consists of a set of mutually exclusive tuples (with overall probability ≤ 1). When needed, we will use G_u to denote the group of tuple u. A *possible world* W of R^p is a subset of tuples of R that respect all the rules in $\mathcal{G}$. The set of all possible worlds of R^p is denoted $\mathcal{W}$.

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Given a (deterministic) relation R whose schema includes a set of numerical attributes $\mathcal{A} = \{A_1, A_2, \ldots, A_d\}$, the *skyline* of R with respect to $\mathcal{A}$, denoted $\text{SKY}(R)$, is the set of *undominated* tuples in R. Assuming that on each attribute higher values are preferable, tuple u dominates tuple v, written $u \succ v$, iff it is $u.A_i \geq v.A_i$ for each $A_i \in \mathcal{A}$ and there exists at least one attribute A_j such that $u.A_j > v.A_j$. Thus: $\text{SKY}(R) = \{u \in R \mid \nexists\, v \in R : v \succ u\}$. When neither $u \succ v$ nor $v \succ u$ hold, we say that u and v are *indifferent*, written $u \sim v$. A *scoring function* $s()$ on the attributes $\mathcal{A}$ is *monotone* iff $u.A_i \geq v.A_i$ $(i = 1, \ldots, d)$ implies $s(u) \geq s(v)$, and is also *domination-preserving* if $u \succ v$ implies $s(u) > s(v)$.

2 The Skyline of a Probabilistic Relation

In [2] we showed how domination among probabilistic tuples (*P-domination* for short) can be defined by using ingredients drawn from order theory. In particular, $\succ$ is a *strict partial order*, i.e., an irreflexive $(\forall u : u \not\succ u)$ and transitive $(\forall u, v, t : u \succ v \wedge v \succ t \Rightarrow u \succ t)$ binary relation. A *linear* order $\succcurlyeq$ is a strict partial order that is also *connected*, i.e., for any two distinct tuples u and v, either $u \succcurlyeq v$ or $v \succcurlyeq u$. A linear order $\succcurlyeq$ is called a *linear extension* of $\succ$ iff $u \succ v \Rightarrow u \succcurlyeq v$, i.e., $\succcurlyeq$ is *compatible* with $\succ$. It is known that any strict partial order $\succ$ equals the intersection of its linear extensions, i.e., $\succ = \bigcap\{\succcurlyeq \mid \succcurlyeq \in \text{EXT}(\succ)\}$, where $\text{EXT}(\succ)$ denote the set of all linear extensions of $\succ$.

We then bring in probability by exploiting the concept of *ranking semantics*. Such semantics, which are used for supporting top-k queries on probabilistic data, define a linear order on the probabilistic tuples of R^p when the (deterministic) tuples of R are ordered according to $\succcurlyeq$ [7, 9, 3]. More precisely, a *probabilistic ranking function* Ψ is a function that, given R^p and a linear order $\succcurlyeq$ on the tuples of R, yields a *probabilistic linear order* $\succcurlyeq_p = \Psi(\succcurlyeq, R^p)$ on the probabilistic tuples of R^p. The actual ranking of the probabilistic tuples is obtained by computing for each tuple u a value $\psi_{\succcurlyeq}(u)$, so that $u \succcurlyeq_p v$ iff $\psi_{\succcurlyeq}(u) > \psi_{\succcurlyeq}(v)$.

The definition of P-domination is then as follows [2]:

Definition 1 (P-domination). *Let $R^p = (R, p, \mathcal{G})$ be a probabilistic relation, and let $\succ$ be the domination relation on the tuples in R when considering the skyline attributes $\mathcal{A}$. Let Ψ be a probabilistic ranking function on R^p. For any two tuples u and v in R^p, we say that u* P-dominates *v, written $u \succ_p v$, iff for each linear extension $\succcurlyeq$ of $\succ$, with associated probabilistic linear order $\succcurlyeq_p = \Psi(\succcurlyeq, R^p)$, it is $u \succcurlyeq_p v$, that is:*

$$\boxed{u \succ_p v \iff u \succcurlyeq_p v, \;\; \forall\, \succcurlyeq_p = \Psi(\succcurlyeq, R^p), \succcurlyeq \in \text{EXT}(\succ)} \tag{1}$$

Consequently, similarly to the deterministic case, the skyline of R^p is defined as:

$$\boxed{\text{SKY}(R^p) = \{u \in R \mid \nexists\, v \in R : v \succ_p u\}} \tag{2}$$

3 P-domination with x-relations

In [2] we have put forward the basic principles that allow P-domination to be checked without materializing any linear extension of $\succ$. In particular, for $u \succ_p v$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

to hold it has to be $\psi_{\succcurlyeq}(u) > \psi_{\succcurlyeq}(v)$ for all linear extensions $\succcurlyeq$ of $\succ$, that is: $u \succ_p v \Leftrightarrow \min_{\succcurlyeq \in \text{EXT}(\succ)} \{\psi_{\succcurlyeq}(u)/\psi_{\succcurlyeq}(v)\} > 1$. Consequently, if one finds a linear order that is the most unfavorable one for u with respect to v, and $\psi_{\succcurlyeq}(u) > \psi_{\succcurlyeq}(v)$ holds for this "extremal" order, then it will necessarily hold for all other orders compatible with $\succ$. This approach amounts to determine a set of *P-domination rules* that can be checked without the need of materializing any linear extension of $\succ$. Regardless of the specific probabilistic ranking function Ψ, the two relevant cases to consider for P-domination rules are:

$\mathbf{u \succ v}$: When u dominates v, $u \succcurlyeq v$ has necessarily to hold; we can thus analyze how other tuples should be arranged in the linear order so as to minimize the ratio $\psi_{\succcurlyeq}(u)/\psi_{\succcurlyeq}(v)$.

$\mathbf{u \not\succ v}$: If u does not dominate v, then extremal linear order corresponds to the case where: 1) $u \succcurlyeq t$ only for those tuples t that u dominates, and 2) $t' \succcurlyeq v$ only for those tuples t' that dominate v.

According to these principles, in [2] we have derived specific P-domination rules for the independent case. In the following we concentrate on the x-relation model and derive P-domination rules for a variety of relevant ranking semantics.

3.1 Expected Rank [3]

Given a linear order $\succcurlyeq$ on the tuples of R, the rank of u in a possible world W is the number of tuples in W that precede u [3]. The expected rank of a tuple u is thus given as the expected value of the rank of u on the set of all possible worlds, each weighted by its probability. This can be expressed as:

$$ER_{\succcurlyeq}(u) = p(u) \times \sum_{t \succcurlyeq u, t \notin G_u} p(t) \; + (1 - p(u)) \times \sum_{t \notin G_u} p(t) \; + \sum_{t \in G_u, t \neq u} p(t)$$

where the first term is the expected rank of u in a possible world in which u appears, whereas the other terms account for the contribution of those worlds that do not contain u (where u has rank $|W|$).

Let $H_{G,\succcurlyeq}(u) = \sum_{t \succcurlyeq u, t \notin G_u} p(t)$ be the overall probability of those tuples not in G_u that are better than u according to $\succcurlyeq$. Further, let $P = \sum_{t \in R^p} p(t)$ be the overall probability of all tuples, $P_{u,v} = P - p(u) - p(v)$, and define $P_{G_u} = \sum_{t \in G_u, t \neq u} p(t)$, as the probability of all tuples but u in G_u.

According to Definition 1, and setting $\psi_{\succcurlyeq}(u) = -ER_{\succcurlyeq}(u)$, tuple u P-dominates v iff it is $\max_{\succcurlyeq \in \text{EXT}(\succ)}\{ER_{\succcurlyeq}(u)/ER_{\succcurlyeq}(v)\} < 1$, i.e., $ER_{\succcurlyeq}(u) < ER_{\succcurlyeq}(v)$ for any $\succcurlyeq \in \text{EXT}(\succ)$. This is equivalent to:

$$u \succ_p v \Leftrightarrow \boxed{\frac{p(u)}{p(v)} > \max_{\succcurlyeq \in \text{EXT}(\succ)} \left\{ \frac{P_{u,v} + 1 - H_{G,\succcurlyeq}(v) - P_{G_v}}{P_{u,v} + 1 - H_{G,\succcurlyeq}(u) - P_{G_u}} \right\}} \tag{3}$$

The two cases to consider for checking the validity of Inequality 3 are dealt with as follows. For both here we describe the general case in which u and v belong to different groups first, leaving out for lack of space the (easier) case $G_u = G_v$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

$\mathbf{u \succ v}$: The first rule we derive is sufficient (but not necessary) for P-domination to occur, and has the advantage that it can be more efficiently checked than the other we derive for the case $u \succ v$.

Since $u \succ v$, it is $H_{G,\succcurlyeq}(v) \geq H_{G,\succcurlyeq}(u) + p(u) - P_{G_v}$, which is obtained by assuming a worst-case scenario for u in which all tuples $t \in G_v$ other than v dominate u. Substituting in Inequality 3 it is obtained:[1]

$$\boxed{u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq 1 \ \wedge \ \frac{p(u)}{P_{G_u}} \geq 1} \qquad \text{(ER:Rule 1)}$$

In case Rule 1 is not satisfied, a preliminary key observation is that, for any linear order $\succcurlyeq$ that extends $\succ$, it is $H_{G,\succcurlyeq}(u) \in [H^-_G(u), H^+_G(u)]$, where the two bounds are respectively defined as:

$$H^-_G(u) = \sum_{t \succ u, t \notin G_u} p(t) \qquad H^+_G(u) = \sum_{u \not\succ t, t \notin G_u} p(t)$$

Clearly, $H^-_G(u)$ is the best possible case for u, since only those tuples from other groups that dominate u are also better than u according to $\succcurlyeq$, whereas the worst case for u arises when u is better only of those tuples of other groups that it dominates.

Now, the ratio in the right-hand side of Inequality 3 can be maximized by adequately arranging those tuples t that are indifferent to either u or v (or both). Since $u \succ v$, and $\succ$ is transitive, only three cases can occur:

$\mathbf{t \sim u, t \succ v}$: In order to favor v with respect to u it has necessarily to be $t \succcurlyeq u$ whenever $t \notin G_u$. Let $IB(u,v) = \sum_{t \sim u, t \succ v, t \notin G_u} p(t)$ stand for the mass of probability of such tuples.

$\mathbf{t \sim v, u \succ t}$: For the same reason it has to be $v \succcurlyeq t$.

$\mathbf{t \sim u, t \sim v}$: To analyze this case, first observe that setting $t \succcurlyeq u \succcurlyeq v$ when $t \in G_v$ would penalize only u. Let $II_{G_v}(u,v) = \sum_{t \sim u, t \sim v, t \in G_v} p(t)$ be the total mass of probability of such tuples. At this point the ratio in Inequality 3 would be written as:

$$\frac{N(u,v)}{D(u,v)} \stackrel{\text{def}}{=} \frac{P_{u,v} + 1 - H^-_G(v) - P_{G_v}}{P_{u,v} + 1 - H^-_G(u) - IB(u,v) - II_{G_v}(u,v) - P_{G_u}} \qquad (4)$$

Consider now those tuples $t \notin G_u \cup G_v$, whose total mass of probability is $\Delta(u,v) = \sum_{t \sim u, t \sim v, t \notin G_u, t \notin G_v} p(t)$. The two alternatives to consider are either $t \succcurlyeq u \succcurlyeq v$ or $u \succcurlyeq v \succcurlyeq t$. The choice actually depends on the value of $N(u,v)/D(u,v)$: If $N(u,v) < D(u,v)$, then setting $t \succcurlyeq u \succcurlyeq v$ would lead to the ratio $(N(u,v) - \Delta(u,v))/(D(u,v) - \Delta(u,v))$, which is *lower* than $N(u,v)/D(u,v)$. On the other hand, when $N(u,v) > D(u,v)$, then $(N(u,v) - \Delta(u,v))/(D(u,v) - \Delta(u,v)) > N(u,v)/D(u,v)$, thus the case $t \succcurlyeq u \succcurlyeq v$ is the one to consider. Finally, when $N(u,v) = D(u,v)$ Inequality 3 degenerates to $p(u) > p(v)$.

[1] Note that in case neither inequality holds strictly in Rule 1 it is still safe to say that $u \succ_p v$, since we assume a domination-preserving tie-breaking rule.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Putting all together, the 2nd P-domination rule is:

$$u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq \begin{cases} \dfrac{N(u,v)}{D(u,v)} & \text{if } \dfrac{N(u,v)}{D(u,v)} \leq 1 \\ \dfrac{N(u,v) - \Delta(u,v)}{D(u,v) - \Delta(u,v)} & \text{otherwise} \end{cases} \qquad \text{(ER:Rule 2)}$$

$u \not\succ v$: When u does not dominate v, P-domination can still occur. In this case it is immediate to see that the ratio in Inequality 3 is maximized by setting $H_{G,\gtrdot}(v) = H^-_G(v)$ and $H_{G,\gtrdot}(u) = H^+_G(u)$, thus:

$$u \not\succ v \ \wedge \ \frac{p(u)}{p(v)} > \frac{P_{u,v} + 1 - H^-_G(v) - P_{G_v}}{P_{u,v} + 1 - H^+_G(u) - P_{G_u}} \qquad \text{(ER:Rule 3)}$$

3.2 Expected Score [3]

In [3], the case where tuples are ranked by their *expected score*, $ES(u) = s(u) \times p(u)$, is also considered. Here, the scoring function $s()$ is assumed to be domination-preserving, i.e., $u \succ v$ implies $s(u) > s(v)$. For this ranking semantics (the only one that explicitly depends on tuples' scores) it is immediate to derive that:

$$u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq 1 \qquad \text{(ES:Rule 1)}$$

This holds since, for any scoring function $s()$, it has to be $s(u) \times p(u) > s(v) \times p(v)$, i.e., $p(u)/p(v) > s(v)/s(u)$. If $u \not\succ v$, then for any given value of the ratio $p(u)/p(v)$ the right-hand side can be made arbitrarily large and P-domination does not hold. If $u \succ v$, it is $s(v)/s(u) = 1 - \epsilon$, with $\epsilon > 0$ arbitrarily small, from which the result follows.

It is remarkable that, although ranking by expected scores is highly dependent on actual score values, the skyline of R^p is, even in this case, insensitive to attribute values. A major pitfall of expected score ranking is that it completely ignores correlation among tuples, and this obviously propagates to skylines. Further, since the above is the *only* P-domination rule, the skyline obtained from the expected score semantics will be always larger than the skyline in the deterministic case (since $u \succ v$ is necessary for P-domination to hold).

3.3 U-Topk [7], U-kRanks [7], and Global-Topk [9]

The semantics we consider in this section do not define a ranking in the proper sense of term, since ranking varies with the value of k. Therefore, our results do apparently not apply to them. However, P-domination can still be defined by considering the case of top-1 queries. This holds since the skyline is the union of top-1 answers, thus we can limit the analysis to the case $k = 1$, and report as $\textsc{Sky}(R^p)$ those tuples that, for at least one linear order $\gtrdot$ compatible with $\succ$, are the top-ranked ones.

For these semantics, which for lack of space we cannot describe here, we can provide a unified analysis, due to the following observation (see also [8]):

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Observation 1 *For any probabilistic relation R^p, whose tuples are linearly ordered according to $\succcurlyeq$, the result of a top-1 query computed according to the U-Top1, U-1Ranks, and Global-Top1 semantics is the same, and is given by the tuple u that maximizes the* top-1 probability*, which for x-relations is:*

$$\Pr_{\succcurlyeq}^{top1}(u) = p(u) \times \prod_{G \neq G_u} \left(1 - \sum_{t \in G, t \succcurlyeq u} p(t)\right) \tag{5}$$

Thus, for such semantics one can set $\psi_{\succcurlyeq}(u) = \Pr_{\succcurlyeq}^{top1}(u)$. Now, by defining $Q_{G,\succcurlyeq}(u) = \prod_{G \neq G_u} \left(1 - \sum_{t \succcurlyeq u, t \in G} p(t)\right)$, it is obtained:

$$u \succ_p v \Leftrightarrow \boxed{\frac{p(u)}{p(v)} > \max_{\succcurlyeq \in \text{EXT}(\succ)} \left\{ \frac{Q_{G,\succcurlyeq}(v)}{Q_{G,\succcurlyeq}(u)} \right\}} \tag{6}$$

$u \succ v$: To derive the first rule we can assume a worst case scenario for u where all tuples of G_v other than v dominate u, obtaining $Q_{G,\succcurlyeq}(v) \leq Q_{G,\succcurlyeq}(u) \frac{1-p(u)}{1-P_{G_v}}$. This, substituted into Equation 6, gives us the following rule:

$$\boxed{u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq \frac{1-p(u)}{1-P_{G_v}}} \qquad \text{(Top1:Rule 1)}$$

If Rule 1 does not hold, we can still bound $Q_{G,\succcurlyeq}(u)$ by observing that the best possible order for u is the one where u is preceded by only those tuples that dominate u, while the worst order is the one that puts u after all tuples not dominated by u. Thus, it is $Q_{G,\succcurlyeq}(u) \in [Q_G^-(u), Q_G^+(u)]$, where the bounds are defined as follows:

$$Q_G^-(u) = \prod_{G \neq G_u} \left(1 - \sum_{u \not\succ t, t \in G} p(t)\right) \qquad Q_G^+(u) = \prod_{G \neq G_u} \left(1 - \sum_{t \succ u, t \in G} p(t)\right)$$

As in Section 3.1, in order to maximize the ratio in the right side of Equation 6 one should only arrange those tuples t that are indifferent to u or v (or both). If u is to be penalized, we should maximize the number of tuples preceding it. To this end, the tuples $t \notin G_u, t \sim u, t \succ v$ and $t \in G_v, t \sim u, t \sim v$ can be all ordered so as to precede u, since they either precede also v or do not contribute to its top-1 probability. Consider now those tuples $t \notin G_v : t \sim u, t \sim v$. The two alternatives to consider are either $t \succcurlyeq u \succcurlyeq v$ or $u \succcurlyeq v \succcurlyeq t$. To this end we first note that, since $u \succ v$, it is $\sum_{t \succcurlyeq u} p(t) \leq \sum_{t \succcurlyeq v} p(t)$, thus having $t \succcurlyeq u \succcurlyeq v$ would actually lower the ratio $(1 - \sum_{t \succcurlyeq u, t \notin G_u \cup G_v} p(t))/(1 - \sum_{t \succcurlyeq v, t \notin G_u \cup G_v} p(t))$, leading to an overall lower value of the right side of Equation 6. By defining $B\_IB(u, v)$ as:

$$B\_IB(u,v) \stackrel{\text{def}}{=} \left(1 - \sum_{\substack{(t \succ u \vee (t \sim u \wedge t \succ v) \vee \\ (t \sim u \wedge t \sim v)), t \in G_v}} p(t)\right) \times \prod_{\substack{G \neq G_u \\ G \neq G_v}} \left(1 - \sum_{\substack{(t \succ u \vee (t \sim u \wedge t \succ v)) \\ t \in G}} p(t)\right)$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

the second P-domination rule is thus obtained as:

$$\boxed{u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq \frac{Q_G^+(v)}{B\_IB(u,v)}} \qquad \text{(Top1:Rule 2)}$$

u $\not\succ$ v: It is easy to see that, since now it could happen that $v \succcurlyeq u$, it is:

$$\boxed{u \not\succ v \ \wedge \ \frac{p(u)}{p(v)} > \frac{Q_G^+(v)}{Q_G^-(u)}} \qquad \text{(Top1:Rule 3)}$$

4 Complexity Analysis

In this section we analyze, for the alterative ranking semantics covered in Section 3, the time complexity for computing the skyline of a probabilistic relation R^p consisting of N tuples. We first cover the simpler case of the expected scores semantics. In this scenario, the only P-domination rule (ES:Rule 1) is a constant time, $\mathcal{O}(1)$, operation. Consequently, the skyline of R^p can be computed in $\mathcal{O}(N^2)$ time.

Turning to consider the expected rank semantics, it is important to first characterize the time complexity of the three P-domination rules described in Section 3.1:

Rule 1: In the worst case, the evaluation over all pairs of tuples requires $\mathcal{O}(N^2)$ time. This clearly holds when the probability of each group is known in advance, but it remains true even if group probabilities need to be determined (since this can be done before starting to compare tuples).

Rule 3: A single check on a pair of tuples apparently has a complexity of $\mathcal{O}(N)$, because of $H_G^-(v)$ and $H_G^+(u)$; however, such values do not depend on the specific pair of tuples being compared, and thus can be pre-computed for each tuple, which collectively requires $\mathcal{O}(N^2)$ time. Thus, a single evaluation is a constant time, $\mathcal{O}(1)$, operation and evaluation on all pairs of tuples requires $\mathcal{O}(N^2)$ time.

Rule 2: A single evaluation requires $\mathcal{O}(N)$ time, because of $IB(u,v)$, $II_{G_v}(u,v)$, and, possibly, $\Delta(u,v)$. Unless one can afford a $\mathcal{O}(N^2)$ storage overhead for book-keeping (which we do not consider here), such quantities cannot be computed in advance because they depend on both u and v.

Therefore, because of Rule 2, computing the skyline of R^p under the expected rank semantics requires $\mathcal{O}(N^3)$ time in the worst case. This result also applies to the three semantics in Section 3.3 by replacing, in the above arguments, $H_G^-(v)$ and $H_G^+(u)$ with $Q_G^-(u)$ and $Q_G^+(u)$ in Rule 3, and $IB(u,v)$, $II_{G_v}(u,v)$, $\Delta(u,v)$ with $B\_IB(u,v)$ in Rule 2. Since Rule 2 is the most expensive one, it is obvious that efficient algorithms for computing the skyline of a probabilistic relation should postpone Rule 2 as much as possible, so as to check it the minimum number of times.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

5 Final Discussion

It is known [5], that different ranking semantics yield quite different results for top-k queries. We show through an example that this is also the case when such semantics are used for skyline queries. Consider a traffic-monitoring application that collects data by means of a set of radars, a sample of last-hour recording being shown in Figure 1. Because of radar locations, a same car cannot be detected by two radars within an interval of one hour. Each radar reading has associated a Prob value, representing the confidence one has in the reading.

TID	Plate No	Radar	Time	Speed	Prob
t_1	X-123	L1	10:53	90	0.2
t_2	W-246	L2	10:50	100	0.15
t_3	W-246	L3	10:40	95	0.1
t_4	Z-456	L1	10:32	110	0.1
t_5	Z-456	L2	10:30	130	0.3
t_6	H-121	L3	10:30	110	0.2
t_7	Y-324	L4	10:30	90	0.5
t_8	X-827	L4	10:20	105	0.35
t_9	X-827	L5	10:15	90	0.4
t_{10}	C-442	L5	10:10	120	0.3
t_{11}	C-442	L2	10:05	140	0.1

Ranking semantics	$\text{SKY}(R^p)$
Expected Rank	$\{t_5, t_7\}$
Expected Score	$\{t_1, t_2, t_4, t_5, t_7, t_8, t_{11}\}$
U-Top1, U-1Ranks, Global-Top1	$\{t_1, t_5\}$

Fig. 1. A probabilistic relation (left) and skylines (rights)

A skyline query on the Time and Speed attributes finds those readings that are the most recent ones and concern high-speed cars. In the deterministic case it is $\text{SKY}(R) = \{t_1, t_2, t_4, t_5, t_{11}\}$, and Figure 1 (right) shows the skylines of R^p for the different ranking semantics. Although t_5 is part of all skylines, this is not the case for other tuples (e.g., t_7). As in [5], we argue that there is not a "best" ranking semantics, rather the choice might depend on the specific application at hand and user preferences. Understanding the properties of the different semantics is therefore a, both practical and theoretical, interesting research issue.

References

1. Agrawal, P., Benjelloun, O., Sarma, A.D., Hayworth, C., Nabar, S.U., Sugihara, T., and Widom, J.: Trio: A System for Data, Uncertainty, and Lineage. In: VLDB 2006. pp. 1151–1154. Seoul, Korea (Sep 2006)
2. Bartolini, I., Ciaccia, P., Patella, M.: Getting the Best from Uncertain Data. In: SEBD 2011. pp. 9–20. Maratea, Italy (Jun 2011)
3. Cormode, G., Li, F., Yi, K.: Semantics of Ranking Queries for Probabilistic Data and Expected Ranks. In: ICDE 2009. pp. 305–316. Shanghai, China (Apr 2009)
4. Dalvi, N.N. and Suciu, D.: Efficient Query Evaluation on Probabilistic Databases. In: VLDB 2004. pp. 864–875. Toronto, ON (Aug 2004)
5. Li, J., Saha, B., Deshpande, A.: A Unified Approach to Ranking in Probabilistic Databases. In: VLDB 2009. pp. 502–513. Lyon, France (Aug 2009)
6. Sarma, A.D., Benjelloun, O., Halevy, A.Y., Widom, J.: Working Models for Uncertain Data. In: ICDE 2006. Atlanta, GA (Apr 2006)
7. Soliman, M.A., Ilyas, I.F., Chang, K.C.C.: Top-k Query Processing in Uncertain Databases. In: ICDE 2007. pp. 896–905. Istanbul, Turkey (Apr 2007)
8. Yan, D. and Ng, W.: Robust Ranking of Uncertain Data. In: DASFAA 2011. pp. 254–268. Hong Kong, China (Apr 2011)
9. Zhang, X., Chomicki, J.: On the Semantics and Evaluation of Top-k Queries in Probabilistic Databases. In: DBRank 2008. pp. 556–563. Cancun, Mexico (Apr 2008)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Relaxed Queries over Data Streams*

Barbara Catania, Giovanna Guerrini, Maria Teresa Pinto, and Paola Podestà

Università di Genova, Italy

Abstract. Relaxation skyline queries have been proposed, in the relational context, as a solution to the so-called empty answer problem. Given a query composed of selection and join operations, a relaxation skyline query relies on the usage of a relaxation function (usually, a numeric function) to quantify the distance of each tuple (pair of tuples in case of join) from the specified conditions and uses a skyline-based semantics to compute the answer. Though the empty answer problem is extremely relevant also in a streaming context, where users may not be acquainted with the actual data arriving on the stream, it has been largely neglected. Specifically, no solutions have been proposed so far for skyline-based relaxation over data streams. In this paper, we define relaxation skyline queries for window-based join over data streams, propose one processing algorithm and present a preliminary experimental evaluation of the designed technique.

1 Introduction

The last decade has been characterized by the raise of data intensive applications, with new data querying needs, and novel processing environments. Data integration applications, Web services, sensor networks, data stream management systems, P2P, cloud computing, and hosting are only few examples of these emerging technologies. The novel processing requirements related to these new contexts made traditional query processing approaches unsatisfactory and required the development of specific advanced query processing techniques.

One specific issue for such advanced techniques concerns *approximation*. Indeed, data characteristics (e.g., heterogeneity, incompleteness, and uncertainty), resource limitations, huge data volumes, and volatility, typical of emerging applications and technologies, suggest it may be preferred to relax the query definition, using Query Relaxation techniques, or to generate an approximate result set, with quality guarantees, using Approximate Query Processing techniques, instead of getting an unsatisfactory answer. An answer can be unsatisfactory because either the user has to wait too long for getting the result, or the answer is empty, or the answer contains too many tuples.

In this paper we are interested in Query Relaxation techniques for solving the empty answer problem. Two different approaches have been provided to address this issue: the first approach relies on techniques for rewriting the query using weaker conditions, in order to get a larger answer set [8, 10, 15]; the second approach exploits quantitative or qualitative preferences in order to relax the query

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

and returning the best results, leading to the definition of top-k [7] and skyline queries [3]. Skyline queries rely on qualitative preferences and determine best results in terms of a partial relation among items, defined as a dominance relation with respect to a set of given attributes (representing the user preference), by returning those items that are not dominated by any other item (*skyline items*).[1] With respect to the second group of approaches, *relaxation skyline* (r-skyline) queries have been proposed in the relational context with the aim of using some system-defined preferences for relaxing selection and join conditions and avoid the empty answer problem [9]. The basic idea of r-skyline queries is to use a relaxing function (usually, a numeric function) to quantify the distance of each tuple (pair of tuples in case of join) from the specified conditions and to rely on a skyline-based semantics to compute the results. The relaxed evaluation of the query thus provides a non-empty answer while being close to the original query formulated by the user. Relaxation skyline queries have been proposed for stored relational data [9] and a similar approach has been provided for sensor networks [11].

While the empty answer problem has been deeply investigated for stored data, few proposals exist for data stream management. In this context, as discussed in [5], several approximation approaches have been proposed to deal with limited or constrained resource availability during data stream processing. However, only few approximation techniques finalized at improving the quality of result, either in terms of completeness or accuracy, have been defined. Such techniques would however be very useful in the streaming context, where the limited knowledge of the users about the actual data arriving on the stream may often lead to the execution of queries returning an unsatisfactory answer, e.g., queries that return an empty result over several windows.

This paper addresses this problem by presenting some preliminary results of an on-going research concerning the definition of relaxation queries and related processing techniques for data streams. In particular, the contribution of the paper concerns the definition of r-skyline queries for window-based join over data streams and a related processing algorithm. The proposed algorithm has been obtained by integrating the approaches presented in [12], for processing skyline queries over a single data stream, and in [9], for processing r-skylines for stored relational data, and extending them to deal with r-skylines over selection and window-based join queries. A preliminary experimental evaluation of the designed technique is also presented, showing the impact of relaxation on performance.

The paper is organized as follows. Section 2 introduces basic concepts on data streams and presents r-skyline queries. A processing algorithm for r-skylines is then presented in Section 3 while preliminary experimental results are reported in Section 4. Some concluding remarks about the obtained results and next steps we plan to follow in our research are then provided in Section 5.

[1] Given a set of points, each corresponding to a list of values for the relevant attributes, a point A dominates a point B if it is better in at least one dimension and equal or better in all the others, with respect to some ordering [3].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2 Relaxation skyline queries for data streams

A data stream is a continuous, unbounded, and potentially infinite sequence of data (tuples, in this paper). In a data stream, each item is associated with a timestamp, either assigned by the source dataset or by the system at arrival time. Queries over data streams can be either one-time, if they are evaluated once on a given subset of data, or continuous, if they are continuously evaluated as soon as new data arrive. According to STREAM [1], continuous queries can be evaluated over data streams and time-varying relations. A time-varying relation is a function that maps each time value to a set of tuples, representing the relation state (i.e., a classical relation) at a certain time instant. Continuous queries are evaluated at each time instant on the relation states and on the subsets of the data streams available at that instant. Window operators are applied on streams in order to compute, at each time instant, a subset of the items arrived so far. Windows can be either time-based, if all the tuples arrived in the last k time units are retained, or count-based, in case the last k tuples are retained.

Example 1. Consider an e-commerce application, dealing with two data streams, `Order` and `Delivery`, containing tuples concerning customer orders (orderID, customer, cost, dateOrder) and product delivery (orderID, clerk, dateOrder). We are interested in determining, in a continuous way, information about orders, whose cost is equal to 500 Euro, related to deliveries performed in the last 3 minutes. The corresponding STREAM's CQL [2] query is:

```
SELECT o.orderID, o.customer
FROM Order o [unbounded], Delivery d [RANGE 3 Minutes]
WHERE o.cost = 500 and o.orderID = d.orderID
```

In the query, `[unbounded]` and `[RANGE 3 Minutes]` are two window operators. At each time instant, the first one returns the set of tuples arrived from the beginning of the stream up to now; the second one returns the set of tuples arrived in the last 3 minutes. ◊

The concept of relaxation skyline (r-skyline), first introduced in [9] for stored data, extends the concept of skyline query [3] to deal with derived attributes, each representing the distance of the considered tuple (pair of tuples, in case of join) to a condition contained in the query. In the following, we consider queries corresponding to a binary join over two data streams, followed by a number of selections. Distances between tuples and query conditions can be computed using a relaxation function defined as follows.

Definition 1 *Let S and T be two data streams. Let $C_1 = S.A_i\ \theta\ v$ and $C_2 = S.A_i\ \theta\ T.A_j$, where θ is a comparison operator, A_i is an attribute of S, A_j is an attribute of T, and v is an allowed value for attribute A_i. Let s be a tuple in S and t a tuple in T. Function RELAX over (s, C_1) returns 0 if $s.A_i\ \theta\ v$, $|s.A_i - v|$ otherwise. Function RELAX over (s, t, C_2) returns 0 if $s.A_i\ \theta\ t.A_j$, $|s.A_i - t.A_j|$ otherwise. In the following, $RELAX(s, t, Q)$, $s \in S$, $t \in T$ denotes the values of function RELAX computed for each condition in Q with respect to tuples s and t.* □

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

tuple	orderID	customer	cost	dateOrder	τ
o_1	0001	James	600	07/11/2010	1
o_2	0002	Thomas	150	07/11/2010	2
o_3	0003	Thomas	300	07/11/2010	3
o_4	0004	Nick	650	07/11/2010	4
o_5	0005	Nick	500	09/11/2010	5
o_6	0006	Nick	100	09/11/2010	6
o_7	0007	James	300	10/11/2010	7
...	...	...	...	...	...

tuple	orderID	clerk	dataOrder	τ
d_1	0001	Simon	08/11/2010	3
d_2	0002	Jack	13/11/2010	4
d_3	0003	Jack	13/11/2010	5
d_4	0004	Simon	14/11/2010	6
d_5	0005	Simon	14/11/2010	7
d_6	0006	Simon	14/11/2010	8
d_7	0007	Jack	15/11/2010	9
...	...	...	...	...

Table 1. Data streams `Order` (left) and `Delivery` (right)

The following definition adapts the definition of dominance and relaxation skyline given in [9] to data streams.

Definition 2 *Let Q be a query, S and T be two data streams, s_1 and s_2 be tuples of S, t_1 and t_2 be tuples of T, w_1 and w_2 be two window operators. The pair of tuples $\langle s1, t1\rangle$ dominates ($\preceq$) the pair $\langle s2, t2\rangle$ if: (i) relaxed values in $RELAX(s_1, t_1, Q)$ are lower than or equal to all the corresponding relaxed values in $RELAX(s_2, t_2, Q)$; (ii) at least one relaxed value in $RELAX(s_1, t_1, Q)$ is lower than the corresponding relaxed value in $RELAX(s_2, t_2, Q)$.*

The r-skyline of S and T with respect to Q, w_1 and w_2, denoted as $rs(S, T, Q, w_1, w_2)$, is the relation that at time τ contains the tuples in $(S[w_1] \times T[w_2])(\tau)$ that are not dominated by any tuple in $(S[w_1] \times T[w_2])(\tau)$. □

Example 2. Consider Example 1 and the relations in Table 1, representing a portion of data streams `Order` and `Delivery`. Timestamps τ are progressive integer numbers, corresponding to the arrival minute. When applying the relaxation function $RELAX$ to condition $C \equiv$ `Orders.cost` $=$ `500` and tuples o_1, o_2, o_3, the following values are obtained: $RELAX(o_1,C_1) = 100, RELAX(o_2,C_1) = 350, RELAX(o_3,C_1) = 200$. Now consider the query of Example 1. Relaxing only condition C (since the join condition relies on order identifiers, which cannot be approximated in a relevant way), r-skyline computation is performed as follows:

- at times $\tau = 1$, $\tau = 2$, the result set is empty (no join);
- at $\tau = 3$, join execution returns just one pair $\langle o_1, d_1\rangle$, which is in turn returned as result of the r-skyline at time 3;
- at $\tau = 4$, the matching pairs are $\langle o_1, d_1\rangle$ and $\langle o_2, d_2\rangle$; based on Definition 2, $\langle o_1, d_1\rangle \preceq \langle o_2, d_2\rangle$ and $\langle o_1, d_1\rangle$ is returned as result of the r-skyline at time 4;
- at $\tau = 5$, the matching pairs are $\langle o_2, d_2\rangle$ and $\langle o_3, d_3\rangle$, tuple d_1 has already expired; based on Definition 2, $\langle o_3, d_3\rangle \preceq \langle o_2, d_2\rangle$ and $\langle o_3, d_3\rangle$ is returned as result of the r-skyline at time 5. ◊

3 Relaxation skyline processing over window-based joins

In this section, we describe a processing algorithm (denoted by NRJL - Non Relaxed Join Lazy) for r-skyline queries over data streams. The proposed algorithm is obtained by merging the Pruning Join algorithm presented in [9] for

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

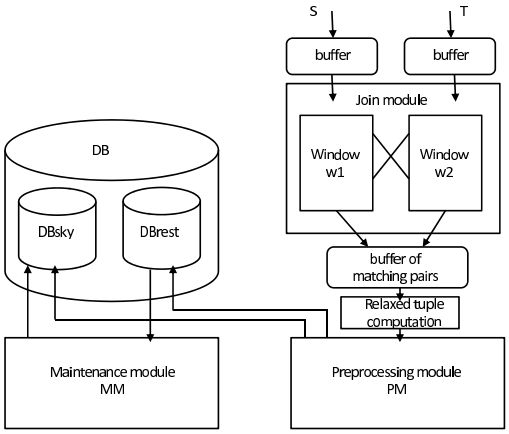


Fig. 1. NRJL Reference architecture

relational data with the Lazy method presented in [12], for skyline computation over data streams. In the following, we assume that all conditions in the query are relaxed except join conditions. This approach is useful, for instance, when join conditions rely on automatically generated semantic-less identifiers.

Fig. 1 shows the architecture at the basis of our technique. To achieve independence on the tuple input rate, arriving tuples are placed in an input buffer (an alternative solution would be that of discarding some input tuples, relying on some load shedding approach [13], or summarizing arrived tuples through synopses [4]). The architecture can be divided into two parts. The first part executes the window-based join of the input streams, S and T, considering two arbitrary windows, w_1 and w_2, one for each stream. Any join algorithm can be employed. In the following, we assume that *symmetric hash join* [14] is employed. This algorithm keeps a hash table for each data stream, indexing the tuples in the current window. Each time a new tuple arrives, it is added to the corresponding hash table and *probed* against the other hash table. Pairs of tuples generated by the join module are stored in a buffer, in ascending order of their arrival time. Arrival and expire times of pairs of matching tuples depend on the arrival and expire time of the composing tuples. More precisely, if the join algorithm generates the pair $\langle s, t\rangle$ and a^s, a^t, w_1, w_2 are the arrival times and windows of S and T, respectively, then the arrival time of the pair is $a = \max(a^s, a^t)$ while the expiration time of the pair is $e = \min(a^s + w_1, a^t + w_2)$.

After computing the join, the second part of the architecture deals with r-skyline computations over window-based join results. Skyline computation is realized through two main modules: a *pre-processing module* (PM) and a *maintenance module* (MM). Module PM is activated as soon as a new pair of tuples $\langle s, t\rangle$ is returned by the join module. It performs two main tasks: (i) relaxing selection conditions by generating one d-dimensional tuple u, corresponding to RELAX(s, t, Q) (see Definition 1); (ii) storing u inside a database DB, partitioned into DB_{sky} and DB_{rest}, which store tuples that are and are not in the current skyline, respectively. These last tuples are maintained since they may

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Query	W	400	800	1600	3200	6400
Q_2	**uniform**	2	2	3	3	7
Q_2	**anticorrelated**	2	7	11	11	10
Q_4	**uniform**	7	7	12	10	8
Q_4	**anticorrelated**	16	19	21	21	22

Table 2. DB_{sky} average sizes corresponding to varying window sizes

become skyline points after some skyline point expires (see [12] for details). More precisely, if u is dominated by tuples in DB_{sky}, u is inserted in DB_{rest}, otherwise tuples dominated by u in DB_{sky} are dropped and u is inserted in DB_{sky} as a new skyline point.

Module MM handles expiring tuples. It is invoked each time a skyline tuple u in DB_{sky} expires. As first step, MM drops u from DB_{sky}. At this point, some tuples in DB_{rest} may become new skyline tuples. To perform such computations, MM, as a second step, drops all expired tuples from DB_{rest} and determines the tuples in DB_{rest} dominated by u. For each of such tuples r', it checks whether other tuples in DB_{sky} exist dominating r'. r' is added to set U only if no other tuple in DB_{sky} dominates it. Finally, the skyline of set U is computed, to find the tuples to be moved from DB_{rest} to DB_{sky}.

In order to efficiently performs search operations against repositories, we assume relaxed tuples inside DB_{sky} and DB_{rest} are indexed using R-trees [6].

4 Preliminary experimental results

The goal of the experimental evaluation is to investigate the impact of relaxation on performance. To this aim, we consider: (i) processing time of individual tuples; (ii) amortized processing time, computed as the sum of processing times of individual tuples, divided by the number of processed tuples.

In the experimental evaluation, data streams consists of tuples with three attributes. Each attribute has the interval $[0, 1]$ as domain. Experiments have been executed on queries composed by a join and several selections (equality checks with 0). We suppose that the time-based window is recomputed upon each new tuple arrival. Windows of size 400, 800, 1600, 3200, 6400 seconds have been considered. Due to the presence of buffers, the experimental evaluation does not depend on the tuple arrival frequency; however, we considered a tuple arrival every 5 seconds. Assuming an alternate arrival between the two streams, this means that each stream produces a tuple every 10 seconds.[2] Finally, we assumed that each stream contained 30 windows, resulting in a minimal stream size of 1200 tuples and a maximal stream size of 19200 tuples.

Input data have been generated through an automatic generator, following the techniques proposed in [3], using two distinct distributions: *uniform* and

[2] We remark that, due to the join computation, the tuple arrival frequency for modules PM and MM will be much higher and, every second, it will correspond to the average number of matching pairs (about 20 in the performed experiments).

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

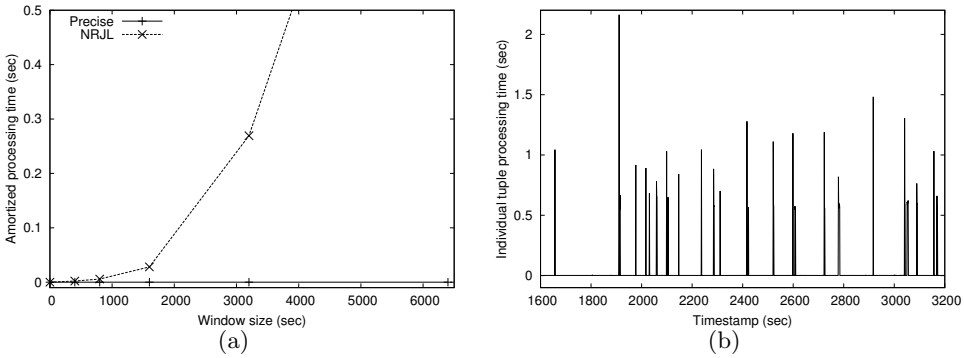


Fig. 2. (a) Amortized cost at varying window size for uniform distribution; (b) Individual tuple processing time for uniform distribution

anticorrelated. Skyline size and thus of processing time [12] are known to be badly influenced by anticorrelated distributions. In our context, since the skyline is computed on join result, an anticorrelated distribution could be imposed on join result, and not only on the inputs. This experiment is left as future work.

Two query types have been considered: *Q4*, containing one selection condition for each attribute and all them are relaxed (for a total of four relaxed conditions), and *Q2*, containing one selection condition for each attribute and only one of them is relaxed for each stream (for a total of two relaxed conditions).

Exp. 1. The average size of the r-skyline in each time instant, corresponding to the average size of DB_{sky}, is determined. The size grows in the number of relaxed conditions: each query type corresponds to a skyline computation on tuples of a given dimension and the skyline size is known to grow in the dimension of the input tuples. Table 2 shows the average dimension of DB_{sky} depending on data distribution and window size. The skyline points for the uniform distribution are always fewer than or equal to those for the anticorrelated distribution.

Exp. 2. Amortized time of NRJL is compared with that of the precise (i.e., without condition relaxation) query. Precise queries are executed as follows: as soon as a new tuple r arrives, we check whether r meets the selection conditions stated for the stream. If all such conditions are met, the symmetric hash join is executed, obtaining the tuples matching r. Fig. 2(a) shows the variation in the average amortized time (in seconds) for queries executed over the various window sizes, for the uniform distribution (results for anticorrelated distribution are similar). The precise query processing time is considerably lower than the time required for relaxed execution. In the precise processing, indeed, no auxiliary data structures, need to be maintained and no relaxation is applied.

Exp. 3. The processing time for each single tuple, starting from the time instant at which the join window becomes full, is analyzed. Fig. 2(b), represents the average individual tuple processing time for NRJL. All processing times are not null, even if, due to a scale problem, they look as they were. Peaks represent the activations of the MM module. Similar results were obtained for the anticorrelated distribution.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

5 Discussion and conclusion

An ongoing work concerning the processing of relaxed queries over data streams has been presented. The concept of relaxation skyline has been extended to data streams and a processing algorithm has been described. Preliminary experimental results show that, as expected, relaxing queries penalizes performance. Performance can be improved in at least two ways, currently under investigation. One approach consists in designing more efficient algorithms for relaxation skyline computation. Algorithms anticipating some skyline computation during window-based join execution, or reducing the size of the maintained state information, should be investigated. The second direction is to adopt an adaptive processing approach to switch from precise queries to skyline-based ones as soon as, based on some dynamically monitored QoD parameters, the system understands that this is needed for improving result quality. A switch from a skyline-based computation to a precise one will occur as soon as parameters indicate that a precise computation can generate a satisfactory result.

References

1. A. Arasu et al. STREAM: The Stanford Stream Data Manager. *IEEE Data Eng. Bull.*, 26(1):19–26, 2003.
2. S. Babu and J. Widom. Continuous Queries over Data Stream. *SIGMOD Record*, 30(3):109–120, 2001.
3. S. Börzsönyi, D. Kossmann, and K. Stocker. The Skyline Operator. In *ICDE*, pages 421–430, 2001.
4. F. Buccafurri and G. Lax. Approximating Sliding Windows by Cyclic Tree-like Histograms for Efficient Range Queries. *Data Knowl. Eng.*, 69(9):979–997, 2010.
5. B. Catania and G. Guerrini. Approximate Queries with Adaptive Processing. In B. Catania and L. Jain, editors, *Advanced Query Processing - Issues and Trends*. Springer, 2012.
6. A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *SIGMOD Conference*, pages 47–57, 1984.
7. I. F. Ilyas, G. Beskales, and M. A. Soliman. A Survey of Top-k Query Processing Techniques in Relational Database Systems. *ACM Comput. Surv.*, 40(4), 2008.
8. A. Kadlag et al. Supporting Exploratory Queries in Databases. In *DASFAA*, pages 594–605, 2004.
9. N. Koudas et al. Relaxing Join and Selection Queries. In *VLDB*, 2006.
10. C. Mishra and N. Koudas. Interactive Query Refinement. In *EDBT*, 2009.
11. L. Pan, J. Luo, and J. Li. Probing Queries in Wireless Sensor Networks. In *ICDCS*, pages 546–553, 2008.
12. Y. Tao and D. Papadias. Maintaining Sliding Window Skylines on Data Streams. *IEEE Trans. Knowl. Data Eng.*, 18(2):377–391, 2006.
13. N. Tatbul et al. Load Shedding in a Data Stream Manager. In *VLDB*, pages 309–320, 2003.
14. A. N. Wilschut and P. M. G. Apers. Dataflow Query Execution in a Parallel Main-Memory Environment. In *PDIS*, pages 68–77, 1991.
15. X. Zhou et al. Query Relaxation using Malleable Schemas. In *SIGMOD Conference*, pages 545–556, 2007.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A Logic-based Language for Data Streams*

Carlo Zaniolo

University of California at Los Angeles
zaniolo@cs.ucla.edu

Abstract. Data Stream Management Systems (DSMS) have attracted much interest, and various extensions of relational-database query languages have been proposed for data streams. However, relational query languages were built on the solid bedrock of logic, while current DSMS languages and their computation models are missing such foundations. In this paper, we show that continuous queries can be characterized using the familiar concepts of closed-world and local stratification, leading to Streamlog that allows a freer and more natural usage of nonmonotonic constructs than Datalog. Thus, Streamlog takes the query languages of DSMS to new levels of expressive power and removes the limitations that severely impair current commercial systems and research prototypes.

1 Introduction

Data stream management systems represent a vibrant area of new technology for which researchers have extended database query languages to support continuous queries on data streams [2, 1, 5, 7, 12, 4, 15]. These database-inspired approaches have produced remarkable systems and applications, but have yet to deliver solid theoretical foundations for DSMS data models and query languages—particularly if we compare to those of relational and deductive databases that delivered concepts and models of great power and elegance [8, 16, 17]. Thus in this paper, we show that logic provides a natural formalism and simple solutions for many of the difficult problems besetting DSMS: by using concepts such as Reiter's Closed World assumption [14] and local stratification[13] we achieve a natural and efficient support of nonmonotonic constructs in recursive rules.

This extendded abstract is organized as follows. In the next section, we present a short discussion of related work and then, in Section 3, we explore the problem of supporting order and recursion on single stream queries for both monotonic and non-monotonic constructs. Thus, in Section 4, we introduce Streamlog, which is basically Datalog with modified well-formedness rules for negation. These rules guarantee both simple declarative semantics and efficient execution (Section 5). Because of possible skews between their timestamps, multiple streams pose complex challenges at the logical and implementation levels. We propose a solution for this problem in Section 6.

2 Continuous Queries in DSMS

Data streams can be modeled as append-only relations on which the DSMS supports continuous queries [2]. As soon as tuples arrive in the input stream, the

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

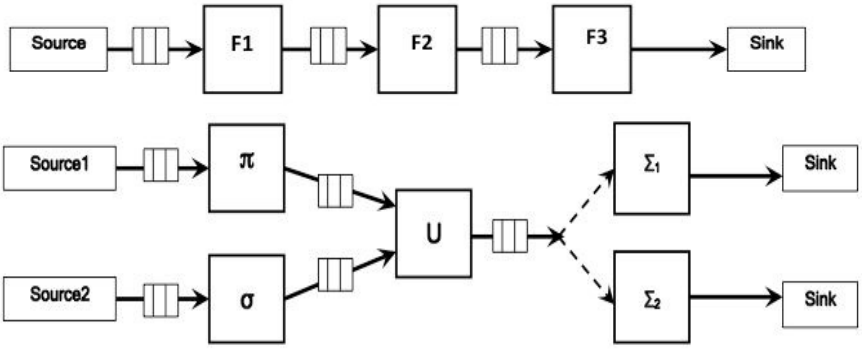


Fig. 1. Continuous Query Graphs

DSMS is expected to decide, in real time or quasi real-time, which additional results belong to the query answer and promptly append them to the output stream. In this incremental computation model no output can be taken back; therefore, the DSMS might have to delay producing a tuple until it is sure that the tuple belongs to the final output—a certainty that for many queries is only reached after the DSMS has seen the whole input. The queries showing this behavior, and operators causing it, are called *blocking*, and have been characterized in [2] as follows: "A blocking query operator is one that is unable to produce the first tuple of the output until it has seen the entire input." Clearly, blocking query operators are incompatible with the computation model of DSMS and should be disallowed, whereas all non-blocking queries should instead be allowed.

The main previous result on blocking queries is that non-monotonic query operators are blocking, whereas monotonic operators are non-blocking [10, 9]. Given that negation and traditional aggregates are non-monotonic, most current DSMS simply disallow them in queries, although this exclusion causes major losses in expressive power [11]. However, in this paper we present a more sophisticated analysis suggesting that these losses are avoidable, since (i) the partial orderings used in [10, 9] are not the same as the subset ordering used in databases and Horn clauses, and (ii) previous research has made great strides in coping with non-monotonicity via concepts such as stratification and stable models.

Queries on data streams are commonly visualized using workflow models such as those in Figure 1, that show the pipelined execution used by the DSMS for continuous queries. The boxes labelled `Source` at the left of our graph, depict tuples coming from an external stream source or a database relation. For instance in the first query, the source feeds incoming tuples to a buffer; then query operator `F1` takes the tuples from this buffer and feeds them to its output buffer that supplies operator `F2`, and so on. As shown in Figure 1, some boxes might consist of very simple operators, e.g., the relational algebra operators of projection, selection and union. In general, however, the boxes can implement much more complex functions, including pattern search operators or data mining functions [15]written in procedural languages, or other languages. Here will assume that those boxes consist of Streamlog rules.

A key assumption is that operators are order-preserving. Thus each operator takes tuples from the front of its input queue and adds any tuple(s) it produces to the tail of its output buffer. Thus, since the operators denoted by boxes in

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

the query graph are defined by Streamlog rules, the semantics of our continuous query is defined by the logic program consisting of (i) the goal defined by the `Sink` node (ii) the rules in the boxes leading to such a goal, and (iii) the facts streaming from the `source` nodes into the rules in our boxes. We assume that our data streams are explicitly timestamped , since the first column of our tuples is a timestamp that either (i) was created by the external device that created the tuple (external timestamp) or (ii) it was added by the DSMS at the time it received the tuple (internal timestamp). In either case, tuples are arranged and processed by increasing values of their timestamps.

3 Single Stream Processing

Let us first consider the example of a single input stream of messages of the form `msg(Time, MsgCode)` and say that we are looking for repeated occurrences of a given message with code "red". Then the following Datalog rule can be used to describe multiple occurrences of the same alarm code "`X`":

Example 1. Repeated occurrences of the same alarm.

$$\texttt{repeated}(\texttt{T}, \texttt{X}) \leftarrow \texttt{msg}(\texttt{T}, \texttt{X}), \texttt{msg}(\texttt{T0}, \texttt{X}), \texttt{T} > \texttt{T0}.$$

Then the query ?`repeated`(`T`, `red`) could be used to signal the repeated occurrences of code "red," which, e.g., might be used by an application to sound an alarm. Thus, our alarm is triggered for all but the first occurrence of code red.

The semantics of query Q on a stream, such as `msg`, is defined by the cumulative answer that Q has returned until time τ. This cumulative answer at time τ must be equal to the answer computed upon the database containing all the data stream tuples with timestamp $\leq \tau$. In a blocking query, this equality only holds at the end of the input, whereas for a continuous non-blocking query it must hold for every instant in time.

Massive data streams can easily exceed the system storage capacity. In DSMS, this problem is addressed with windows or other synopses, which can be easily expressedin Streamlog. But, unlike in some DSMS [1], windows do not play a key role in the semantics of Streamlog.

The Importance of Order. Since query operators return sequences of tuples that are fed into the next query operator, assuring the correct order of their output sequences becomes critical. To illustrate this point, say that we modify Example 1, above, by keeping the body of the rule unchanged; but then we change the head of the rule so that the timestamp of the former occurrence is used, rather than the current one:

Example 2. Time-warped repetitions ?`wrepeated`(`Time`, `X`)

$$\texttt{wrepeated}(\texttt{T0}, \texttt{X}) \leftarrow \texttt{msg}(\texttt{T}, \texttt{X}), \texttt{msg}(\texttt{T0}, \texttt{X}), \texttt{T} > \texttt{T0}.$$

We immediately realize that there is a problem, since repetitions normally arrive in an order that is different from that of their previous occurrences. For instance, we might have that a message with code α arrives at time t_α, followed

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

by a message with code β, which is then repeated immediately, while the first repetition of α arrives 10 minutes later. Then, to produce tuples by increasing timestamps, we will need to hold up the output for 10 minutes. In the worst case, the delay required can be unbound, although punctuation marks and windows can be used to alleviate the problem in many situations. Although the situation of *unbound wait* has not been studied in the literature, it is clear that in many cases it can be as bad as that of blocking queries. Therefore, rules such as that of Example 2 must be disallowed, although they contain no negation or other nonmonotonic operators.

3.1 Negated Goals

The addition of order-inducing constraints in the rules offers unexpected major benefits when dealing with negated goals. Say that we want to detect the first occurrence of "code red" warning. For that, we only need to make sure that once we receive such a message there is no identical other message preceding it:

Example 3. First occurrence of code red: ?$\texttt{first}(\texttt{T}, \texttt{red})$.

$$\texttt{first}(\texttt{T}, \texttt{X}) \leftarrow \quad \texttt{msg}(\texttt{T}, \texttt{X}), \neg\texttt{previous}(\texttt{T}, \texttt{X}).$$
$$\texttt{previous}(\texttt{T}, \texttt{X}) \leftarrow \texttt{msg}(\texttt{T0}, \texttt{X}), \texttt{T0} < \texttt{T}.$$

These queries only use negation on events that, according to their timestamps, are past events. Thus the queries can be answered in the present: they are non-blocking. Therefore, they should be allowed by a DSMS compiler, which must therefore be able to set them apart from other queries with negation which are instead blocking.

For instance, a blocking query is the following one that finds the last occurrence of code-red alert:

Example 4. Last occurrence of code red: ?$\texttt{last}(\texttt{T}, \texttt{red})$.

$$\texttt{last}(\texttt{T}, \texttt{Z}) \leftarrow \texttt{msg}(\texttt{T}, \texttt{Z}), \neg\texttt{next}(\texttt{T}, \texttt{Z}).$$
$$\texttt{next}(\texttt{T}, \texttt{Z}) \leftarrow \texttt{msg}(\texttt{T1}, \texttt{Z}), \texttt{T1} > \texttt{T}.$$

This is obviously a blocking query, inasmuch as we do not have the information needed to decide whether the current red-alert message is actually the final one, while messages are still arriving. Only when the data stream ends, we can make such an inference: to answer this query correctly, we will have wait till the input stream has completed its arrival, and then we can use the standard CWA to entail the negation that allows us to answer our query. But the standard CWA assumption will not help us to conclude that the query in Example 3 is non-blocking. We will instead exploit the timestamp ordering of the data streams to define a Progressive Closing World Assumption (PCWA) that can be used for that. In our definition, we will also include traditional database facts and rules, since these might also be used by continuous queries. Thus, we consider a world consisting of a regular database facts and one timestamped-ordered stream:

Progressive Closing World Assumption (PCWA): Once a fact $\texttt{stream}(\texttt{T}, \ldots)$ is observed in the input stream, the PCWA allows us to add to our knowledge base $\neg\texttt{stream}(\texttt{T1}, \ldots)$, provided that $\texttt{T1} < \texttt{T}$, and $\texttt{stream}(\texttt{T1}, \ldots)$ is not entailed by our fact base augmented with the stream facts having timestamp $\leq \texttt{T}$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Therefore, our PCWA for a single data stream revises the standard CWA of deductive databases with the provision that the world is in fact expanding according to its timestamps. For entailment, we can use notions of entailment that were shown to preserve consistency in Datalog, including the least models of Horn Clauses and the perfect models of (locally) stratified programs.

4 Streamlog

In Streamlog, base predicates, derived predicates and the query goal are all timestamped in their first arguments. The same safety criteria used in Datalog can be used in Streamlog. Furthermore, we assume that timestamp variables are made safe by equality chains equating their values to the timestamps in the base stream predicates[1].

We can now propose obvious syntactic rules that will avoid blocking behavior in the temporal rules of safe Streamlog programs.

Sequentially Structured Programs.

- *Strictly Sequential:* A rule is said to be *Strictly sequential* when the timestamp of its head is $>$ than every timestamp in the body of the rule. A predicate is strictly sequential when all the rules defining it are strictly sequential.
- *Sequential:* A rule is said to be sequential when it satisfies the following three conditions:
  (i) the timestamp of its head is equal to the timestamp of some positve goal,
  (ii) the timestamp of its head is $>$ or $\geq$ than the timestamps of the remaining goals, and
  (iii) if the rule is recursive, then all negated goals that are recursive with the head predicate are strictly sequential.
- A program is said to be *sequentially structured* when all its rules are sequential or strictly sequential.

The programs in Example 3 is sequentially structured while those in Example 2 and Example 4 are not. Thus we have here a simple and intuitive notion that characterizes non-blocking continuous queries, including those that use negation. Moreover sequentially structured programs are easy for a compiler to recognize, and they are conducive to a very efficient implementation.

This is quite obvious for the program in Examples 3 since this is stratified with respect to negation [17], but it is also true for the program in Example 5 that is not. The program in Example 5 is locally stratified and thus has a unique stable model called the perfect model [17]. Moreover, while recognizing locally stratified programs is in general Π^1_1-complete [6], sequentially structured programs are ease to recognize and implement because of the pattern of their temporal arguments that is akin to XY-stratification [17].

By their ability of pushing negation or aggregates into recursion, sequentially structured Streamlog programs can implement algorithms that could not be expressed or efficiently implemented in Datalog. This is demonstrated by

[1] Expressions such as $T2 = f(T1)$ or $T2 = T1 + 1$ cannot be used to deduce the safety of $T2$. Only equality can be used for timestamp arguments.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Example 5 that solves the well-known shortest path problem. The rules in this example specify the operations discussed next. The last two rules take the arcs with the current timestamp and assign them to path, provided that no shorter path with the same endpoints was computed earlier. The actual recursive computation of `path` is performed by the middle rule. This rule uses $\texttt{lgr}(\texttt{T1}, \texttt{T2}, \texttt{T})$ to select the larger of its first two arguments, whereby `T` will always coincide with the current timestamp as per differential fixpoint used in the implementation of these rules [17]. The negated goal in this rule checks that no shorter path with the same endpoints was computed earlier. The top two rules visit `path` atoms with timestamp `T` so produced and only retain the shortest ones, for a given start point and end point.

Example 5. Continuous shortest paths in graphs defined by stream of arcs.

$$
\begin{array}{ll}
\texttt{minpath}(\texttt{T},\texttt{X},\texttt{Y},\texttt{D}) \leftarrow & \texttt{path}(\texttt{T},\texttt{X},\texttt{Y},\texttt{D}), \neg\texttt{shorterpath}(\texttt{T},\texttt{X},\texttt{Y},\texttt{D}).\\
\texttt{shorterpath}(\texttt{T},\texttt{X},\texttt{Z},\texttt{D}) \leftarrow & \texttt{path}(\texttt{T},\texttt{X},\texttt{Z},\texttt{D1}), \texttt{D1} \leq \texttt{D}.\\
\texttt{path}(\texttt{T},\texttt{X},\texttt{Z},\texttt{D}) \leftarrow & \texttt{path}(\texttt{T1},\texttt{X},\texttt{Y},\texttt{D1}), \texttt{path}(\texttt{T2},\texttt{Y},\texttt{Z},\texttt{D2}),\\
 & \texttt{lgr}(\texttt{T1},\texttt{T2},\texttt{T}), \texttt{D} = \texttt{D1}+\texttt{D2}, \neg\texttt{shorter}(\texttt{T},\texttt{X},\texttt{Z},\texttt{D}).\\
\texttt{path}(\texttt{T},\texttt{X},\texttt{Y},\texttt{D}) \leftarrow & \texttt{arc}(\texttt{T},\texttt{X},\texttt{Y},\texttt{D}), \neg\texttt{shorter}(\texttt{T},\texttt{X},\texttt{Y},\texttt{D}).\\
\texttt{shorter}(\texttt{T},\texttt{X},\texttt{Y},\texttt{D}) \leftarrow & \texttt{arc}(\texttt{T},\_,\_,\_), \texttt{path}(\texttt{T1},\texttt{X},\texttt{Y},\texttt{D1}), \texttt{T1} < \texttt{T}, \texttt{D1} \leq \texttt{D}.
\end{array}
$$

In this program, we have expressed the transitive closure using quadratic rules, since this requires only the memorization of `path` values. In the linear expression of transitive closure both `arc` and `path` would have required memorization.

Therefore, sequentially structured programs support *negation in recursion* by restricting recursive negated goals to previous timestamps. These programs have the formal semantics and efficient implementation discussed next based on their bistate version. The *bistate* version of a program is obtained by observing that for each timestamp value in the head the rule contains (i) goals that have the same timestmap value as the head, and (ii) goals having previous timestamp values. Now, the head and the goals in (i) are renamed with the prefix "new" and the goals in (ii) are renamed with the prefix"old": by this renaming, every sequentially structured program becomes a stratified program and thus amenable to efficient computation:

Theorem 1. *If P is a Sequentially Structured program then: (i) P is locally stratified, and (ii) the unique stable model of P can be computed by repeating, for each timestamp value, the iterated fixpoint computation of its bistate version.*

5 Multiple Streams and Synchronization

A much studied DSMS problem is how to best guarantee that binary query operators, such as unions or joins, generate outputs sorted by increasing timestamp values [3]. To derive a logic-based characterization of this problem, assume that our `msg` stream is in fact built by combining the two message streams `sensr1` and `sensr2`. For stored data, this operation requires a simple disjunction as follows:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Example 6. Disjunction expressing the union of two streams.

$$\texttt{msg}(\texttt{T},\texttt{S}) \leftarrow \texttt{sensr1}(\texttt{T},\texttt{S}).$$
$$\texttt{msg}(\texttt{T},\texttt{S}) \leftarrow \texttt{sensr2}(\texttt{T},\texttt{S}).$$

However, for data streams, even if `sensr1` and `sensr2` are ordered by their timestamps, this disjunction says nothing about the fact that the output should be ordered. Indeed, assuring such an order represents a serious challenge for a DSMS, due to the time-skews that normally occur between different data streams. Thus, for the union in Figure 1, when one of the two input buffers is empty, we cannot take any tuple from the other buffer, until we know what its timestamp value will be. At the logical level, the problem can be solved by the introduction of predicate $\texttt{missing}_\texttt{i}\texttt{i}$ that checks tuples in the other stream:

Example 7. Synchronized Union of Streams.

$$\texttt{msg}(\texttt{T},\texttt{S}) \leftarrow \texttt{sensr1}(\texttt{T},\texttt{S}), \neg\texttt{missing}_2(\texttt{T}).$$
$$\texttt{msg}(\texttt{T},\texttt{S}) \leftarrow \texttt{sensr2}(\texttt{T},\texttt{S}), \neg\texttt{missing}_1(\texttt{T}).$$

Here $\neg\texttt{missing}_2$ ($\neg\texttt{missing}_1$)guarantees that all the `sensr2` (`sensr2`) tuples with timestamp $< \texttt{T}$ have already been added to the output. Now $\texttt{all}_2$ can be expressed by double negation:

$$\texttt{missing}_2(\texttt{T}) \leftarrow \texttt{sensr1}(\texttt{T},\_), \texttt{sensr2}(\texttt{T2},\texttt{S2}), \texttt{T2} < \texttt{T}, \neg\texttt{msg}(\texttt{T2},\texttt{S2}).$$
$$\texttt{missing}_1(\texttt{T}) \leftarrow \texttt{sensr2}(\texttt{T},\_), \texttt{sensr2}(\texttt{T1},\texttt{S1}), \texttt{T1} < \texttt{T}, \neg\texttt{msg}(\texttt{T1},\texttt{S1}).$$

Thus, we have a sequentially structured program,

Example 8. Union by sort-merging unsynchronized data streams.

$$\texttt{msg}(\texttt{T1},\texttt{S1}) \leftarrow \texttt{sensr1}(\texttt{T1},\texttt{S1}), \texttt{sensr2}(\texttt{T2},\_), \texttt{T2} \geq \texttt{T1}.$$
$$\texttt{msg}(\texttt{T2},\texttt{S2}) \leftarrow \texttt{sensr2}(\texttt{T2},\texttt{S2}), \texttt{sensr1}(\texttt{T1},\_), \texttt{T1} \geq \texttt{T2}.$$

These rules are correct but not complete. In fact, in the first rule we have $\texttt{T2} \geq \texttt{T1}$ instead of, $\neg\texttt{missing}_2(\texttt{T})$: now, the first clause implies the latter but not vice-versa. Moreover, with no tuple in one of the two buffers these rules imply that we have to enter an idle-waiting state that is akin to temporary blocking.

From the viewpoint of users, neither the solution in Example 7 nor that in Example 8 are satisfactory. What users instead want is to write the simple rules shown in Example 6 and let the system take care of time-skews. Therefore in Streamlog, we will allow users to work under the *Naive Synchronization Assumption (NSA)*, whereby $\texttt{all}_1$ and $\texttt{all}_2$ are always satisfied (and likewise when we have the union or join of multiple streams). Observe that this assumption allows us to generalize the PCWA to multiple data streams and provides Streamlog users with the benefits of logical simplicity and expressive power that PCWA entails. However with NSA, the system is left with the responsibility of (i) adding implicit synchronization conditions such as $\texttt{all}_1$, $\texttt{all}_2$ to the rules involving multiple data streams, and (ii) supporting them very efficiently using the backtracking techniques discussed in [3].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

6 Conclusion

This paper has brought logic-based foundations and superior expressive power to DSMS languages which, currently, are dreadfully lacking both. By revising the closed-world assumption for timestamped data streams, and introducing the notion of sequentially structured programs, nonmonotonic constructs can be naturally supported in recursive Streamlog programs. As a result, Streamlog programs can achieve higher levels of expressive power and lower computation complexity than stratified Datalog. Extensions of these results to data streams without timestamps and the efficient implementation of Streamlog programs will be discussed in future reports.

References

1. A. Arasu, S. Babu, and J. Widom. Cql: A language for continuous queries over streams and relations. In *DBPL*, pages 1–19, 2003.
2. B. Babcock, S. Babu, M. Datar, R. Motawani, and J. Widom. Models and issues in data stream systems. In *PODS*, 2002.
3. Yijian Bai, Hetal Thakkar, Haixun Wang, and Carlo Zaniolo. Optimizing timestamp management in data stream management systems. In *ICDE*, 2007.
4. D. Carney et al. Monitoring streams - a new class of data management applications. In *VLDB*, Hong Kong, China, 2002.
5. S. Chandrasekaran and M. Franklin. Streaming queries over streaming data. In *VLDB*, 2002.
6. Peter Cholak and Howard A. Blair. The complexity of local stratification. *Fundam. Inform.*, 21(4):333–344, 1994.
7. C. Cranor et al. Gigascope: High performance network monitoring with an sql interface. In *SIGMOD*, page 623. ACM Press, 2002.
8. Hervé Gallaire, Jean-Marie Nicolas, and Jack Minker, editors. *Advances in Data Base Theory, Vol. 1*, Advances in Data Base Theory. Plemum Press, 1981.
9. Yuri Gurevich, Dirk Leinders, and Jan Van den Bussche. A theory of stream queries. In *DBPL*, pages 153–168, 2007.
10. Yan-Nei Law, Haixun Wang, and Carlo Zaniolo. Data models and query language for data streams. In *VLDB*, pages 492–503, 2004.
11. Y. Law, H. Wang, & C. Zaniolo. Relational languages and data models for continuous queries on sequences and data streams. *TODS*, 36:8:1–8:32, June 2011.
12. Sam Madden, et al. Continuously adaptive continuous queries over streams. In *SIGMOD*, pages 49–61, 2002.
13. Teodor C. Przymusinski. Perfect model semantics. In *ICLP/SLP*, 1988.
14. Raymond Reiter. On closed world data bases. In *Logic and Data Bases*, pages 55–76, 1977.
15. Hetal Thakkar et al. Smm: A data stream management system for knowledge discovery. In *ICDE*, pages 757–768, 2011.
16. Jeffrey D. Ullman. *Principles of Database and Knowledge-Base Systems, Volume I.* Computer Science Press, 1988.
17. C. Zaniolo, S. Ceri, C. Faloutsos, R.T. Snodgrass, V. S. Subrahmanian, and R. Zicari. *Advanced Database Systems.* Morgan Kaufmann, 1997.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A Framework for Biological Data Normalization, Interoperability, and Mining for Cancer Microenvironment Analysis*

Michelangelo Ceci[4], Mauro Coluccia[7], Fabio Fumarola[4], Pietro Hiram Guzzi[3], Federica Mandreoli[6], Riccardo Martoglia[6], Elio Masciari[1], Massimo Mecella[2], and Wilma Penzo[5]

[1] ICAR-CNR
[2] La Sapienza University
[3] Magna Graecia University
[4] Dip. di Informatica - University of Bari "Aldo Moro"
[5] University of Bologna
[6] University of Modena-Reggio Emilia
[7] Dip. di Scienze Biomediche ed Oncologia Umana - University of Bari "Aldo Moro"

Abstract. Over the last decade, the advances in the high-throughput omic technologies have given the possibility to profile tumor cells at different levels, fostering the discovery of new biological data and the proliferation of a large number of bio-technological databases. In this paper we describe a framework for enabling the interoperability among different biological data sources and for ultimately supporting expert users in the complex process of extraction, navigation and visualization of the precious knowledge hidden in a such huge quantity of data. In this framework, a key role is played by the Connectivity Map, a databank which relates diseases, physiological processes, and the action of drugs. The system will be used in a pilot study on the Multiple Myeloma (MM).

1 Introduction

The emergence of affordable high-performance computers, and the high throughput omic technologies are the basis of several projects aiming at building new public molecular profile data repositories on clinical cancer and cultured cancer cell lines. Using such resources, bio-medical researchers can i) publish their data and results, and ii) use the in-lab produced and public data to study a drug candidate, a gene or a disease state to verify hypothesis and generate new knowledge.

Major examples of public bio-technological databases and repositories are: the National Center for Biotechnology Information (NCBI) located in United

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

States[8], the European Bioinformatics Institute (EBI) placed in Europe[9], and the DNA Data Bank of Japan (DDBJ)[10]. They host data about genome, proteins, nucleotides, genes, relationships between phenotype and genotype and more than 21 million citations from biomedical literature. Subsequently, other interesting project initiatives have come out, each of which with the goal of providing useful information with respect to a particular viewpoint of complex biological systems. Gene Ontology (GO)[11] focuses on gene product characteristics and gene product annotation data. GO allows users to query and to extract knowledge from the built ontology. The KEGG database[12] stores a collection of online databases dealing with genomes and enzymatic pathways. The DrugBank[13], the KEGG DRUG[14], the ChEBI[15] databases offer different kinds of bioinformatics and cheminformatics resources that combine detailed drug data with comprehensive drug target information. The NCI-60 [14] database offers tools for storing, querying and downloading molecular profile data of 60 diverse human cancer cell lines to screen compounds for anticancer activity.

In addition to the above described ones, a particularly interesting project is the Connectivity Map (hencefort CMap) [11], which is born with the challenge of establishing relationships among diseases, physiological processes, and the action of drugs according to the same language. The CMap provides a solution to this problem by i) describing all biological states (physiological, disease, or induced with a chemical or genetic construct) in terms of genomic signatures, ii) creating a large public database of signatures of drugs and genes, and iii) developing pattern-matching tools to detect similarities among these signatures.

In this paper, we presents a preliminary proposal that exploits the main potentialities of this bio-medical research trend towards a data-centric science by providing a knowledge support to the study of cancer microenvironments. With respect to CMap, the added value of our proposal consists in linking out CMap with the various types of data and partial knowledge stored in different data banks, including those cited above. By performing a comprehensive analysis of databases, data repositories, and ontologies, our ultimate goal is not to replicate existing data, but to design and develop a Web delivery system which:

1. enables the interoperability among the queryable data sources,
2. captures the different kinds of relationships that exist among them,
3. reinforces the cooperation of heterogeneous and distributed data bank sources for the query processing target,
4. supports the users in the complex process of extraction, navigation and visualization of the knowledge hidden in a such huge quantity of data.

[8] http://www.ncbi.nlm.nih.gov/
[9] http://www.ebi.ac.uk/
[10] http://www.ddbj.nig.ac.jp/
[11] http://www.geneontology.org/
[12] http://www.genome.jp/kegg/
[13] http://www.drugbank.ca/
[14] http://www.genome.jp/kegg/drug/
[15] http://www.ebi.ac.uk/chebi/

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

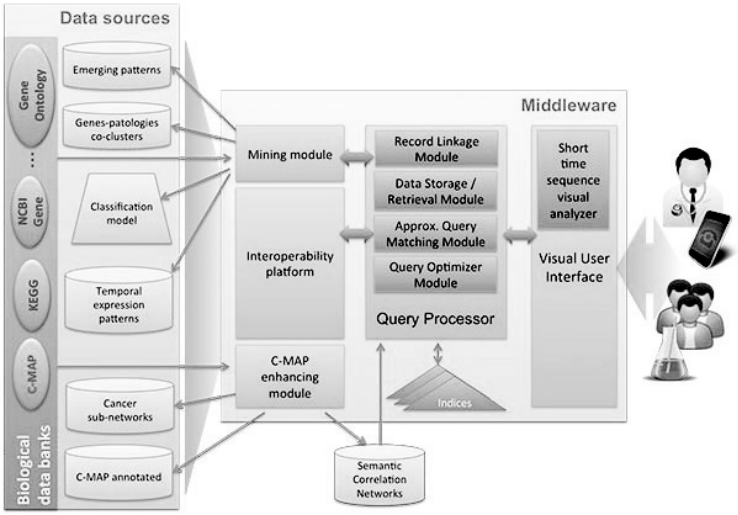


Fig. 1. The system architecture

In particular, to facilitate interoperability (i), we will focus on the normalization problem by creating a semantic layer linking the data sources (ii). On top, innovative algorithms and techniques for querying (iii), mining and visualizing data, models and statistics will enable the extraction of new knowledge (iv).

The main of the Web delivery system will be to assist bio-medical researchers in analyzing tumors microenvironments in order to understand them and identify relationships among tumors, the effect of drugs and the patients' biodiversity. Such relationships are of particular interest for drug repositioning (that is, understanding whether drugs typically used for treating some specific tumors can be used for treating other tumors since they report at a normal state the same genes) and for the identification of novel compounds able to overcome resistance or revert it. The system will be used in a pilot study on the Multiple Myeloma (MM), an incurable malignant plasma cell disease.

2 Our approach

Our goal is to introduce an end-to-end system (whose architecture is depicted in Fig. 1) that would provide a technological support to this issue by exploiting the CMap and additional data sources. In this way, bio-medical researchers will be able to study Cancer microenvironments in order to understand their specificities and the effect of drugs considering the patients' biodiversity. In particular, it is possible to understand the relationships of one tumor with other tumors, to understand mechanisms of drug resistance in tumor Cells to drugs in current use, to elucidate the contribution of the tumor environment in conferring drug resistance, to identify candidate compounds for drug repositioning, to identify a set of candidate gene products for extensive study of their role in drug resistance.

To this end, we will pursue the following objectives:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

- Identification of the data repositories which relate to the CMap;
- Normalization and interoperability of the identified databases and the CMap;
- Use of Data Mining techniques for extracting useful knowledge from data;
- Use of techniques for semantic tagging of the CMap;
- Use of techniques for querying the extended CMap, the identified data repositories and the extracted knowledge as a unique dataspace;
- Use of Visual Query Languages for querying CMap and extracted knowledge.

Our aim is therefore to give answers to the problems that are mandatory for these objectives. In the following, we survey the main problems we foresee in this context and delineate the research directions toward their solution.

2.1 Bio-technological data gathering

Our first goal is to identify the data repositories which can be combined with the Connectivity Map, keeping in mind that the final goal is to allow bio-medical researchers to navigate the stored knowledge as well as to formulate new hypotheses based on the information stored in the bio-technological data repositories.

As a matter of fact, several works in the literature aim at extending the information stored in the CMap [7, 15]. However, they represent ad-hoc extensions. Our goal, instead, is to provide a systematic approach to extend the CMap and make the information it stores interoperable with data available in other biotecnological database such as NCBI (Gene, Geo, Pubmed), ArrayExpress, Gene Ontology, KEGG, and Drug Bank as reported in Fig. 1.

2.2 Data mining and semantic information extraction

By taking into account the requirements coming from bio-medical researchers we will exploit several mining algorithms in order to extract knowledge from the selected datasources. The goal is to better understand tumors and to identify relationships among them, the effect of drugs and the patients' biodiversity.

These algorithms can significantly benefit from the identification of semantic relatedness among entities. In particular, the results of the data mining algorithms can be used to enrich/confirm ontologies that can be used to support semantic-based querying. On the other hand, semantically tagged data can be used to identify relationships to be used in the mining processes.

By exploiting co-clustering approaches, it is possible to discover groups of genes whose gene expressions are simultaneously altered by one or more diseases [2]. To this purpose, hierarchical and non-hierarchical co-clustering techniques can be exploited. Moreover, in order to characterize and describe tumors (or classes of tumors) on the basis of the variability of genomic signatures observed in gene products, network based emerging patterns discovery algorithms [3] can be used. Furthermore, we plan to analyze evolution of diseases through short time series analysis techniques [4]. To this end, both visual data mining and temporal patterns extraction algorithms will be defined. Finally, in order to identify the disease's stage on the basis of expression gene values, collective classification algorithms and ensemble-based algorithms can be exploited. While in the case of collective classification [13] it is possible to handle the autocorrelation (according

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

to which "closer" objects are more related than "furthermost" ones), typically present in data organized in network form (as those considered here, where genes are related to other genes, to diseases, to functional pathways and to miRNAs), in ensemble-based classification [12] different learning models will be combined together (ensemble) in order to define the final model. All the above mentioned mining algorithms will be implemented in the "Mining module" (see Fig. 1).

As regards data correlation analysis, starting from data stored in the CMap, the selected databases and appropriate ontologies, a Semantic Correlation Network will be built. This semantic network will be used to extract sub-networks related to Cancer through the application of network analysis algorithms such as network alignment algorithms [10], clustering [8], and pattern extraction algorithms [6]. The outcomes of this activity will be implemented in the "CMap enhancing module" of the Web delivery system.

2.3 Biotechnological data modeling and management

Once all the data repositories have been identified, additional problems raise. Indeed, the biological data to be analyzed are heterogeneous both in their type and format, since they come from several data sources exhibiting different schema. Moreover, another kind of information that is particularly useful for our goal is the knowledge provided by the mining activities. Again, it differs from the biological databanks not only for the format but mainly for the adopted model as it refers to a mining model rather than operational ones. On the other hand, all the above mentioned data sources are inherently connected, thus the availability of normalization and interoperability solutions that allow analysis tools to deal with information coming from different sources in a unified way is crucial.

In addition, solutions to enrich the CMap with the information gathered from the other biological data sources are necessary to use semantics to search or browse its data. Finally, a flexible query model is necessary that allow stakeholders to query the knowledge in the data sources in a uniform way and to get useful results for analysis purposes. Thus, the main challenges for this goal are:

1. Extension of the CMap with semantic information encoded into ontologies;
2. Normalization and interoperability of the set of data sources;
3. Definition of techniques for effectively and efficiently supporting querying.

As regards the first challenge, RDF annotations to the CMap entities with the support of the selected ontologies will be introduced. The output will be stored in a relational database (CMap annotated, see Fig. 1) containing both entities and functional annotations extracted from ontologies whereas the methods will be implemented in an ad hoc module, called Annotation Module. It will create the first version of CMap annotated, then it will periodically update it by searching for new annotations that can be extracted from publicly available databases.

The second challenge will be dealt with the aim of providing a technological platform to the full interoperability among the selected data banks and the outputs of the mining activities: CMap annotated; the sub-networks related to Cancer; the genes-pathologies co-clusters, the disease (emerging) patterns, temporal expression patterns (extracted from short time sequences) and the disease

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

classification model. As to source participation, the platform will support two alternative options: external sources accessible through Internet querying services and local sources, which the system will have full control and accessibility on.

To this end, the platform will draw inspiration from dataspaces [5]. Indeed, a dataspace follows a data co-existence approach and its main aim is to provide base functionality over all data sources, regardless of how integrated they are, thus shifting the emphasis to a data co-existence. To this end, an ontological language will support the specification of data sources in the form of data schema, mining model or query flow and a mapping language will be used for inter-source mapping specification. As to the latter, the platform will support the gradual specification of schema mapping between sources in a pay-as-you-go fashion [1].

The third challenge is faced through 1) the introduction of a flexible query language and an approximate query matching model that would allow stakeholders to easily query the system and to get useful results, 2) the definition of algorithms and data structures for approximate query answering that would ensure good performances under different system conditions.

The language allows users to specify queries as graphs of biological concepts, biological entities (data instances), predicates on biological entities, and labeled relationships among them. Moreover, it will extend the classical comparison operators with ad-hoc operators to query mined data and models. Query samples that could be specified are "Find all genes that are up-regulated and whose localization is similar to Nucleolus and function is similar to receptor-binding", "Find all thegroups of similar genes whose localization is different from nucleolus that are down-regulated under the effect of drug X", and many others.

Once a query is issued to the system, the query processor module will approximate the query on the dataspace by: 1) defining a query plan that selects the involved sources through the interoperability platform, 2) sending to each selected source the appropriate query, collecting and merging the query results through the application of record linkage techniques [9].

In order to support an efficient query processing on the dataspace, we will study appropriate data structures and algorithms that will support all the different peculiarities of the queried data and of the query language. To this end, two kinds of indices will be concurrently exploited in order to efficiently answer a given query: a) High-level indices; b) Low-level indices (one or more per source). Some of the high-level indices will help in efficiently filtering out unnecessary sources and accessing those having structural and or data properties compatible with the given query only. In this case, ideas from existing successful proposals in different and simpler single-source scenarios will be adapted and exploited (e.g. signatures and bloom filters). Other kinds of high-level indices will instead be helpful when merging answers coming from the queried sources. Some local sources will be equipped with one or more low-level indices, whose kind will depend on the source data. The peculiarities of involved indices and sources will be exploited by algorithms for access plan generation. In case of queries with a very large number of results, the project will study top-k algorithms to maximize the relevance of the results and to minimize the processing time.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2.4 The Web delivery system for the easy-access to the datasources

The Web delivery system will be implemented as a Service Oriented Infrastructure according to which Web-services enabling both access to data and usage of the defined algorithms will be provided. An important feature is the user-friendliness of the whole prototype for potential users. The Web delivery system will then be made accessible by means of a Visual User Interface module that provides biological data experts with a rich user experience during the usage of the tool, both in the querying phase and in the result manipulation phase. The main objective is then the definition of a visual query language specifically targeted to biological data sources analysis and of appropriate visualization techniques.

In order to fully explain the goal we intend to obtain let us consider the query and visualization interface of CMap. By using this tool, a query basically consists in providing a signature file and searching for connected objects. The main difficulties for users are in the text-based syntax of the signature file, which almost requires a kind of programming capabilities, as the syntax should be rigorous. In particular, nowadays the way of writing a query is to create an Excel file and to insert specific values into the columns, according to the given sheet format. Conversely, a graphical interface will be developed, in which the user, through drag&drop of the basic elements needed for building a signature (to be taken from a palette available to the user), is able to visually write such a signature and to use it for querying the system. In the same way, currently the results of the queries are viewed in a table format, and then for each of them a click allows for opening the related specification (again an Excel file). Conversely, a graph-based visualization is envisioned, in which results are shown as nodes of a graph, and the edges represent relationships (e.g., due to sharing of some objects in the structure). Different colors, thickness of the edges, etc. convey specific semantics. A more natural interaction modality will also allow for the use of the interface/tool by users equipped with modern devices, such as tablets, during their normal operations in laboratories.

3 Case study: Study of the Multiple Myeloma

The web framework will be used in a pilot study on the Multiple Myeloma (MM), an incurable malignant plasma cell disease with an incidence of 5 per 100,000 inhabitants, and for that in NCBI GEO are submitted around 6658 samples. MM locates primarily to the bone marrow (BM) in multiple niches that provide a microenvironment which promotes tumor survival. In this context, the main aim of the system will be to allow bio-medical researches to avoid wasting time and funds for the in-vitro verification of potentially meaningless hypothesis by their testing with in silico techniques. Specifically, thanks to the system, it will be possible drive the process of hypothesis generation in: 1) understanding the correlation of the MM with other tumors in terms of gene expressions modifications; 2) defining a characterization of the MM in terms of genes; 3) analyzing the evolution of the pathology; 4) automatically identify MM on the basis of gene expressions modifications and additional information stored in other datasources.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

This analysis might help the drug repositioning task and the identification of novel compounds able to overcome resistance or revert it in the four drugs in current use (Dexamethasone, Bortezomib, High Dose Melphalan, Lenalidomide).

4 Conclusions

In this paper we presented a system for biological data normalization and interoperability devoted to knowledge extraction, data querying and knowledge dissemination for supporting biomedical specialist in the analysis of cancer microenvironments. The system is tailored on the biological data features in order to make it easy to use and provide useful information to the domain experts.

References

1. K. Belhajjame, N. W. Paton, S. M. Embury, A. A. A. Fernandes, and C. Hedeler. Feedback-based annotation, selection and refinement of schema mappings for dataspaces. In *Proc. of EDBT*, pages 573–584, 2010.
2. D.Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. In *ISMB*, pages 145–154, 2002.
3. G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *KDD*, pages 43–52, 1999.
4. J. Ernst and Z. Bar-Joseph. Stem: a tool for the analysis of short time series gene expression data. *BMC Bioinformatics*, 2006.
5. Alon Y. Halevy, Michael J. Franklin, and David Maier. Principles of dataspace systems. In *PODS*, pages 1–9, 2006.
6. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
7. Guanghui Hu and Pankaj Agarwal. Human disease-drug network based on genomic expression profiles. *PLoS ONE*, 4(8):e6536, 08 2009.
8. A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31, September 1999.
9. R. Karmel and D. Gibson. Event-based record linkage in health and aged care services data: a methodological innovation. *BMC Health Services Research*, 2007.
10. O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes, and N. Przulj. Topological network alignment uncovers biological function and phylogeny. *J. of the Royal Society*, 2010.
11. Justin Lamb. The Connectivity Map: a new tool for biomedical research. *Nature Reviews Cancer*, 7(1):54–60, 2007.
12. D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.
13. P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, pages 93–106, 2008.
14. Robert H Shoemaker. The NCI60 human tumour cell line anticancer drug screen. *Nat Rev Cancer*, 6(10):813–823, October 2006.
15. Marina Sirota, Joel T. Dudley, Jeewon Kim, Annie P. Chiang, Alex A. Morgan, Alejandro Sweet-Cordero, Julien Sage, and Atul J. Butte. Discovery and preclinical validation of drug indications using compendia of public gene expression data. *Science Translational Medicine*, 3(96):96ra77, 2011.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A Lightweight Model for Publishing and Sharing Linked Web APIs*

Devis Bianchini, Valeria De Antonellis, and Michele Melchiori

Dept. of Information Engineering – University of Brescia
Via Branze, 38 – 25123 Brescia, Italy
{bianchin|deantone|melchior}@ing.unibs.it

Abstract. The web of Linked Data has been proposed in the last years in order to create a global data graph, that spans data sources, connected by RDF links, and enables the discovery of new resources. Recently, Web APIs have been more and more used to access documents and metadata from the web of Linked Data and to easily compose new applications called web mashups. In this paper, we describe a lightweight model for publishing and sharing Web APIs and mashups on the web of Linked Data. The model has been designed (a) to support providers who publish new Web APIs used for accessing the web of Linked Data, (b) to support the web designer who aims at exploring and selecting available Web APIs for building or maintaining a web mashup, and (c) to make the mashup itself available on top of the web of Linked Data. The functional architecture of a platform based on the model is also introduced.

1 Introduction

The web of Linked Data can be seen as a global database, where resources are identified through URIs, are semantically described and globally connected through RDF links [3]. Recently, Web APIs have been more and more used to access documents and metadata from the web of Linked Data and to easily compose new applications called web mashups. In this paper, we describe a lightweight model for publishing and sharing Web APIs and mashups on the web of Linked Data. A lightweight model for Linked Web APIs built on top of the web of Linked Data has been proposed in [7], where the aim is to support users in the semantic annotation of Web APIs by leveraging on the huge amount of linked web resources. In particular, the SWEET tool [5] has been designed to assist users in annotating HTML descriptions of Web APIs, relying on solutions like Watson [4] for browsing the web of Linked Data, so that it is possible to identify suitable vocabularies (e.g., eCl@ass, Good Relations and FOAF) and use them for the annotation. Our goal in this paper is to extend the model described in [7] to improve the selection from-the-shelf of available Web APIs and to better support their aggregationfor building new web mashups.

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

The motivations of our effort rely on a lack of support for browsing and selecting the right Web APIs from a highly populated repository of third party components. As a motivating example, consider a web designer who wants to make available for his friends a new web application for finding informations, trailers and ratings on Michelangelo Antonioni's movies (see Figure 1, where a screenshot of `MovieGram`[1], a similar application, is shown). To this aim, the web designer has to find and wire Web APIs that provide information about movies (e.g., `Rottentomatoes.com`) and Web APIs to show videos and trailers (e.g., `YouTube`), instead of designing the application from scratch (that could be a non-trivial and time consuming task). On the other hand, browsing and selection of the right functionalities from a huge list of heterogeneous Web APIs is not an easy task if manually performed. To mention the well-known ProgrammableWeb API repository, which registers over 5,600 Web APIs (a number that is continuously growing), the web designer may perform a keyword-based search and find up to 32 Web APIs related to the keyword `movie` and up to 141 APIs for displaying videos, each of them presents several invocable operations with I/Os that must be properly wired (for instance, in the application in Figure 1, videos related to the Antonioni's Zabriskie Point movie are shown on the `YouTube` Web API). Therefore, the web designer must be properly supported in the selection and aggregation of available Web APIs.

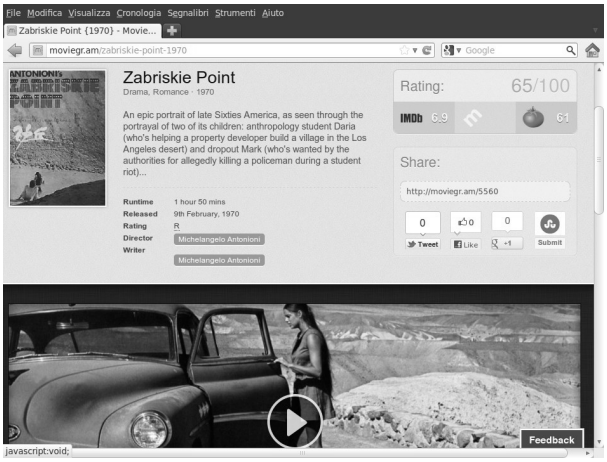


Fig. 1. An example of mashup that mixes up few APIs out of an highly populated repository.

To this aim, we propose the model described in this paper, where: (1) HTTP URIs are used to identify Web APIs, (2) useful (RDF) information are provided on the Web API description when someone looks up a Web API through its URI,

[1] `http://moviegr.am/`.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

and (3) proper links are set to relate Web APIs to each other, thus enabling easy development and sharing of new web applications.

The paper is organized as follows. In Section 2 the lightweight model is described. Section 3 introduces the I-MASH platform, an on-going project, based on the model, that is being designed (a) to support providers who publish new Web APIs used for accessing the web of Linked Data, (b) to support the web designer who aims at exploring and selecting available Web APIs for building or maintaining a web mashup, and (c) to make the mashup itself available on top of the web of Linked Data. Finally, Section 4 closes the paper.

2 Linked Web API model

Following the lesson learnt by the iServe platform [7], we propose the conceptual model shown in Figure 2. The model aims at connecting the contents of the ProgrammableWeb repository of Web APIs with the web of Linked Data to fasten Web API selection and aggregation, going beyond keyword-based searching facilities offered by the repository, given the huge number of Web APIs among which the web designer must choose with few information to support him. We considered the ProgrammableWeb repository since, to the best of our knowledge, it is the most complete collection of Web APIs and user-defined web mashups.

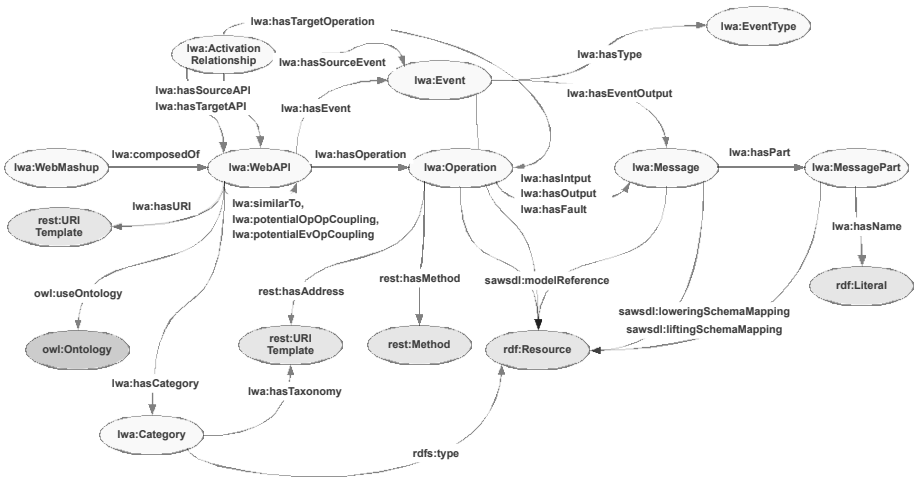


Fig. 2. The semantics-enabled Linked Web API conceptual model.

Basic elements of the model have been defined in the `lwa` namespace. We rely on existing vocabularies, namely SAWSDL, RDF(S) and hRESTS, identified with the `sawsdl`, `rdf`/`rdfs`, `rest` namespace prefixes, respectively. We define a

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

`lwa:WebMashup` as composed of Web APIs. Each Web API is uniquely identified by an URI. Moreover, a set of common elements can be identified in Web API descriptions:

- *inputs* and *outputs*, represented through the `Message` construct, which in turn can be composed of several `MessageParts`;
- *operations*, associated with an `address` (which is built by concatenating the API URI with the operation name), used to invoke the operation itself, and a `method` (e.g., POST and GET HTTP methods) for its invocation;
- *events*, to model user's interactions with the Web API interface (they are commonly used in complex Web APIs such as `GoogleMaps`); events are described by an event type (e.g., `onmouseover`, `onmouseclick`) and the outputs or arguments which are raised on event occurrence (for instance, a click on a map raises an event which contains the coordinates of the points which the mouse is positioned on);
- *categories*, that are taken from public classifications available on the Web (see for example the ProgrammableWeb site, where 67 categories such as `mapping`, `payment`, `search` are considered).

To add semantics to the Linked Web API model, we used the *model reference*, *lifting schema mapping* and *lowering schema mapping* constructs proposed in the SAWSDL specification. Schema mappings are used to provide grouding from the conceptual model to the concrete message formats (lowering schema mapping) and viceversa (lifting schema mapping). Model reference construct is used to associate operation names, inputs and outputs and event arguments to concepts taken from the web of Linked Data.

2.1 Linking Web APIs

Publication of new Web APIs and new web mashups on the web of Linked Data must be properly supported to help providers in identifying links between them. We consider three kinds of links between Web APIs: in the following, for each kind of link, we will provide a short description, an example, semi-automatic techniques relying on the web of Linked Data to support the link identification and the way the link is included in the model in Figure 2. Additional details can be found in [2].

Similar Web APIs

A) Synopsis. Web APIs are linked towards other APIs that provide similar functionalities. This means that linked Web APIs present semantically close I/O messages and operations and compatible categories. Web API similarity is evaluated with respect to a set of operations in the API that is source of the relationship. In Figure 2, Web API similarity is represented through the `lwa:similarTo` construct.
B) Example. Both `Rottentomatoes.com` API and the `Internet Movie Database` (`IMDb`) API present an operation for searching movies; the operation requires as

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

input one or more keywords and returns movie information. Therefore, with respect to this operation, the `Rottentomatoes.com` and `IMDb` APIs are similar.

C) Semi-automatic link identification. We provide a set of matching techniques to identify similar Web APIs, extensively defined in [2]. These techniques are based on the computation of a concept affinity $CAff() \in [0..1]$ to quantify the degree of similarity between pairs of concepts, used in the semantic annotation of, respectively, (i) operations and (ii) I/Os parameters. Here we simply state that $CAff$ is based on both a terminological (domain-independent) matching based on the use of WordNet and a semantic (domain dependent) matching based on the ontologies on the web of Linked Data, used as source of knowledge. The concept affinity evaluation algorithm looks for a path of relationships between two concepts c_1 and c_2; we consider four cases: (1) c_1 and c_2 are defined as equivalent in one of the ontologies; (2) there is a path of *isa* relationships from c_1 to c_2 in one of the ontologies; (3) c_1 and c_2 are not semantically related in any ontology, but their names belong to the same synset in WordNet; (4) c_1 and c_2 are not semantically related in any ontology and their names belong to different synsets in WordNet, but there is a path of hyponymy/hypernymy relations which connects the two synsets. The affinity between two concepts c_1 and c_2 is maximum (that is, equal to 1.0) in the first and third case, otherwise, the highest the length of the path in the second and fourth case, the lowest is concept affinity. Two Web APIs are similar if they are classified in the same category and the average $CAff$ between their operations and I/O messages, namely the *functional similarity degree* between Web APIs, exceedes a threshold experimentally set. The average $CAff$ is computed, through the application of the Dice formula [8], as the average affinity between pairs of concepts, one from the first API and one from the second one. Pairs of concept to be considered in the average similarity computation are selected according to a maximization function that relies on the assignment in bipartite graphs and has been introduced in [1]. This function ensures that each concept from the first API participates in at most one pair with one of the concepts from the second API. The formula the functional similarity degree is summarized in Appendix A.

Op2Op wiring of Web APIs

A) Synopsis. Web APIs can be aggregated by wiring their operations. In this case, the outputs of an operation from the first Web API can be used as inputs of an operation from the second Web API. Potential coupling between Web APIs is checked with respect to a set of operations in the first Web AP. In Figure 2, this kind of link between Web APIs is represented through the `lwa:potentialOpOpCoupling` construct.

B) Example. The `Rottentomatoes.com` API presents an operation to find movies given one or more keywords (for instance, to find Antonioni's movies). Movie titles can be used to retrieve related video by invoking the proper operation from the `YouTube` API.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

C) Semi-automatic link identification. This kind of link is identified by evaluating the average $CAff$ between the outputs of the operations in the first API and the inputs of the operations in the second one. The link is set if the average $CAff$, denoted with *functional Op2Op coupling degree* (see Appendix A), exceeds a threshold experimentally set.

Ev2Op wiring of Web APIs

A) Synopsis. Web APIs can be wired also at the UI level, by coupling an event from the first Web API with an operation from the second one. Event-operation coupling is made according to a publish/subscribe-like mechanism. In this way, interactions with the UI of the first Web API raise events that trigger operations from the second API, ensuring the synchronization of all Web APIs that compose the web mashup. An event-operation pair is represented in the model through an activation relationship. An activation relationship is modeled with reference to the source and target Web APIs (`hasSourceAPI` and `hasTargetAPI` constructs), to the source event (`hasSourceEvent` construct) and to the target operation (`hasTargetOperation` construct). Potential coupling between Web APIs is checked with respect to a set of events in the first Web API. In Figure 2, this kind of link between Web APIs is represented through the `lwa:potentialEvOpCoupling` construct.
B) Example. The `Rottentomatoes.com` API presents an operation to find locations of movie theatres. These locations can be displayed an a map using the `GoogleMaps` APIs. If the zoom level on the map is changed, the list of movie theatres and related information must be properly updated.
C) Semi-automatic link identification. This kind of link can be identified in a similar way with respect to the previous one. The average $CAff$ between the arguments of the events from the first Web API and the inputs of the operations from second one is evaluated and the link is set if the average $CAff$, denoted with *functional Ev2Op coupling degree* (see Appendix A), exceeds a threshold experimentally set.

3 The I-MASH platform

The lightweight Linked Web API model is the basis for the I-MASH platform, an on-going project whose architecture is shown in Figure 3. The core module of the platform is the *Mashup Engine*, which implements the Web API selection and aggregation. The engine also includes the modules for $CAff$ evaluation, relying on the terminological knowledge provided by WordNet and on the domain-specific knowledge from the web of Linked Data, that is accessed through the *Linked Web API Repository*. The *Linked Web API Repository* maintains the representation of Linked Web APIs published according to the model presented in Figure 2. The Linked Web API descriptions refers to things (i.e., URIs of ontological concepts) taken from the web of Linked Data, which the *Linked Web API Repository* is built upon. Other I-MASH modules, such as the *Mashup En-*

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

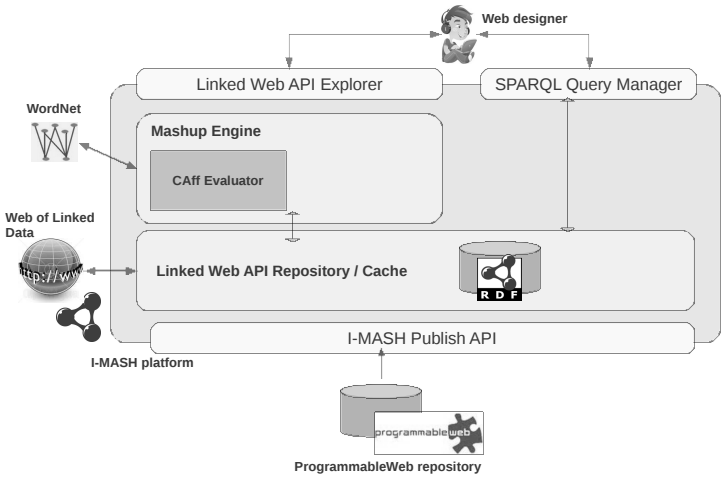


Fig. 3. The I-MASH platform architecture.

gine and the I-MASH *Publish API*, exploit the *Linked Web API Repository* to access the web of Linked Data. The I-MASH *Publish API* module implements the semi-automatic link identification metrics introduced in the previous section and is used to publish new Web APIs extracted from the ProgrammableWeb repository by invoking its methods (`api.programmableweb.com/`). Finally, the *Linked Web API Explorer* supports the web designer to find Web API descriptions and compose them in a web mashup, by leveraging the net of links in the *Linked Web API Repository*. For skilled web designers, we equipped the I-MASH platform with an interface to directly query the *Linked Web API Repository* through the formulation and execution of SPARQL queries. For more details on the I-MASH platform, we refer to [2].

4 Concluding remarks

In this paper we proposed a lightweight model to support both providers who publish new Web APIs used for accessing the web of Linked Data and web designers who aim at exploring and selecting available Web APIs for building or maintaining a web mashup. Additional kinds of relationships to link Web APIs will be studied and analyzed. Among them, we will consider complemetary Web APIs, that is, Web APIs which start from similar inputs and provide alternative functionalities, according to their use in previously defined web mashups. For instance, according to ProgrammableWeb, the `Rottentomatoes.com` and the `YouTube` APIs have been used together in four web mashups out of seven mashups which include the `Rottentomatoes.com` API. The use of traditional support and confidence DM metrics to infer Web API complementarity will be studied, following preliminary results discussed in [6].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

References

1. D. Bianchini, V. De Antonellis, and M. Melchiori. Flexible Semantic-based Service Matchmaking and Discovery. *World Wide Web Journal*, 11(2):227–251, 2008.
2. D. Bianchini and V. De Antonellis. *Linked Data Services and Semantics-enabled Mashup*, chapter on Semantic Search on the Web, pages 281–305. Springer, 2012.
3. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story so Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
4. M. d'Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, and D. Guidi. Toward a New Generation of Semantic Web Applications. *IEEE Inte*, 23(3):20–28, 2008.
5. M. Maleshkova, C. Pedrinaci, and J. Domingue. Semantic annotation of Web APIs with SWEET. In *Proc. of the 6th Workshop on Scripting and Development for the Semantic Web*, 2010.
6. M. Melchiori. Hybrid techniques for Web APIs recommendation. In *Proceedings of the 1st International Workshop on Linked Web Data Management*, pages 17–23, 2011.
7. C. Pedrinaci, D. Liu, M. Maleshkova, D. Lambert, J. Kopecky, and J. Domingue. iServe: a Linked Services Publishing Platform. In *Proceedings of ESWC Ontology Repositories and Editors for the Semantic Web*, 2010.
8. C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

A Formulas for Web API linking identification

Functional Web API similarity

$$Sim_{IO}(\mathcal{W}_i, \mathcal{W}_j) = \frac{1}{3}\left[\frac{\sum_{s,t} CAff(in_s, in_t)}{|IN(\mathcal{W}_j)|} + \frac{\sum_{h,k} CAff(out_h, out_k)}{|OUT(\mathcal{W}_i)|} + \frac{\sum_{l,m} CAff(op_l, op_m)}{|OP(\mathcal{W}_i)|}\right]$$

- $IN(\mathcal{W}_j)$ (resp., $OUT(\mathcal{W}_i)$) is the set of concepts used to annotate the operation inputs of the $\mathcal{W}_j$ Web API description (resp., the operation outputs of the $\mathcal{W}_i$ Web API description);
- $OP(\mathcal{W}_i)$ is the set of concepts used to annotate the operations of the $\mathcal{W}_i$ Web API description;
- $in_s \in IN(\mathcal{W}_i)$, $in_t \in IN(\mathcal{W}_j)$, $out_h \in OUT(\mathcal{W}_i)$, $out_k \in OUT(\mathcal{W}_j)$, $op_l \in OP(\mathcal{W}_i)$, $op_m \in OP(\mathcal{W}_j)$.

Web API Ev2Op coupling

$$Coupl_{EV2OP}(\mathcal{W}_i, \mathcal{W}_j) = \frac{\sum_{s,t} Coupl_{EvOp}(ev_s, op_t)}{|EV(\mathcal{W}_i)|} \quad ev_s \in EV(\mathcal{W}_i),\ op_t \in OP(\mathcal{W}_j)$$

$$Coupl_{EvOp}(ev_s, op_t) = \frac{\sum_{h,k} CAff(out_h, in_k)}{|OUT_{ev}(ev_s)|}$$

Web API Op2Op coupling

$$Coupl_{OP2OP}(\mathcal{W}_i, \mathcal{W}_j) = \frac{\sum_{s,t} Coupl_{OpOp}(op_s, op_t)}{|OP(\mathcal{W}_i)|} \quad op_s \in OP(\mathcal{W}_i),\ op_t \in OP(\mathcal{W}_j)$$

$$Coupl_{OpOp}(op_s, op_t) = \frac{\sum_{h,k} CAff(out_h, in_k)}{|OUT(op_s)|}$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A Framework for Building Multimedia Ontologies from Web Information Sources*

Angelo Chianese, Vincenzo Moscato, Fabio Persia,
Antonio Picariello, and Carlo Sansone

DIS - University of Naples, via Cluadio 21, 80125, Napoli, Italy
{angchian,vmoscato,fabio.persia,picus,carlosan}@unina.it

Abstract. The definition of ontologies within the multimedia domain still remains a challenging task, due to the complexity of multimedia data and the related knowledge. In this paper, we present a novel framework (MOWIS) aiming at realizing a system for building Multimedia Ontologies from Web Information Sources that has been realized within the PRIN 2007-2009 project Cooperare and presented in previous works. In particular, we propose: i) a multimedia ontology model that combines both low level descriptors and high level semantic concepts; ii) automatic construction of ontologies using the FLICKR web services that provide images, tags, keywords and sometimes useful annotation describing both the image content and personal interesting information. Eventually, we describe an example of automatic ontology generation in a specific domain and present some preliminary experimental results.

1 Introduction

The Web 2.0 has changed the relation between users and Internet and nowadays, a lot of repositories containing both multimedia and the related annotations or metadata are publicly available on the web. The main idea beyond this work is that such a kind of information can be efficiently used for an automatic extraction of multimedia knowledge, particularly suitable for a variety of applications, such as multimedia information indexing and retrieval, multimedia content visualization and browsing, learning, reasoning and so on. Indeed, information in the most diffused multimedia databases on the Web (e.g. Flickr, YouTube, etc...) is described by means of "flat" metadata, the most of the times using a predefined set of metadata, or sometimes using small annotations in natural languages: such a kind of structures are substantially inadequate to support complete retrieval by content of multimedia documents.

On the other side, it is well known in the literature that despite the tons of papers produced about multimedia databases and knowledge representation, there is not yet an accepted solution to the problem of *how to represent, organize and manage multimedia data and the related semantics by means of a formal framework*. It is the authors' opinion that there is still a great work to do with respect to the *intensional aspects* of a *multimedia ontology*: (i) *What a multimedia ontology is*: is it a taxonomy, or a semantic network of metadata (tags, annotations)? (ii) *Does a multimedia ontology support concrete data*: what is the role of rough data – image, video, audio data– if any? *What a multimedia semantics is*: how to define and capture the semantics of multimedia data?

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

How to build extensional ontologies: once defined a suitable formal framework, can we automatically build the defined multimedia ontologies?

Throughout the rest of paper, we will try to give an answer to all the previous cited aspects; in particular the original contribution of this work is: (i) to propose a novel multimedia ontology framework, (MOWIS) especially in the image domain; (ii) to propose a technique for building ontologies, that operates on large corpora of human annotated repositories, namely the *FLICKR* database and the related *folksonomy* [8], considering both low level image processing strategies and keywords and annotations produced by humans when they store the produced data.

In the most common vision, *Multimedia Ontologies* represent a way to formally specify the knowledge related to a specific domain by means of the use of multimedia documents, such as images, videos, audio and texts. In particular, they are able to model a domain knowledge exploiting low-level features, structure, semantic concepts of multimedia data and the relationships (of different kinds) among them. Usually low-level features are machine-oriented and can be automatically extracted (as for *MPEG7* descriptors), while semantic concepts are manually provided and are meaningful information only in specific domains. Multimedia ontologies should allow the *mapping* between low-level and high-level information of multimedia data or their parts, thus supporting a more effective retrieval [7]. Respect to the problem of the *semantic annotation* by means of a multimedia ontology that was largely investigated in the literature [13], the automatic building of multimedia ontologies still remains an open issue and a challenging task. Generally, the process for building multimedia ontology is structured into three steps: (i) selection of the concepts to be included in the ontology, (ii) definition of properties and relationships for the concepts, (iii) population and maintenance of the ontology. To this aim, the main approaches for ontology building in the literature are those *concept-driven* and *data-driven*. In the first case the ontology is built by the knowledge related to a specific domain, while in the second case it is directly obtained from multimedia information, but a preliminary domain knowledge is used to select suitable data.

In the following we briefly describe the most diffused techniques in the literature for managing multimedia ontologies from which our work takes its roots. In [2] a first interesting technique for extracting semantic concepts by clustering images on the base of their visual and text features from annotated repositories is illustrated. While, a first proposal of how to combine visual and semantic descriptors to support annotation of multimedia contents in a specific domain is described in [4]. More recently, Bertini et al. presented in [3] the framework *MOM* (*Multimedia Ontology Manager*) based on the concept of *enriched ontology* for automatic video annotation and retrieval. Videos are grouped into clusters on the base of visual features and linked to specific semantic concepts (those corresponding to the objects that are representatives of the clusters). Moreover, facilities for expressing multimedia complex queries on the ontology are provided. In the opposite, *BOEMIE* [11] uses a supervisioned approach based on the extraction of semantic concepts from multimedia data to obtain in a evolutionary and incremental manner a multimedia ontology. Finally, *OntoMedia* [12] is a particular multimedia ontology framework that aims at managing very large collections of information by using techniques of *semantic integration* of metadata.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

In this paper, we first provide an algorithm for creating image ontology in a specific domain gathering high and low level multimedia information. We then give an example of automatic construction of image ontology and a discussion of the encountered problems and the provided solutions. Finally, we conclude showing preliminary experimental results. We want to remark that the system supporting the described framework has been realized within the PRIN 2007-2009 project *Cooperare* and presented in previous works [9, 10].

2 Building an Image Ontology

2.1 An Image Ontology Model

Stressing its conceptual nature, an ontology may be considered as a *theory* used to represent relevant notions about domain modeling in terms of concepts, relationships and constraints on them. Let us consider the image domain: as usual in a given media, we detect symbols, objects and concepts; in a certain image we have regions of pixels (symbol) related to portions of multimedia data; these regions are instances (object) of a certain concept. In other words, we can detect concepts but we are not able to disambiguate among the instances without some specific knowledge. A simplified version of the described vision process will consider only two main levels: *Low* and *High*. In fact, the knowledge associated to an image can be easily described at two different levels of analysis: (i) descriptions of raw images or of their parts(regions); (ii) general or domain-specific image content concepts that can be derivable or less from low-level.

Following such general considerations, an *Image Ontology* can be modeled by a directed and labeled graph $(\mathcal{V}, \mathcal{E}, \rho)$, where: (i) $\mathcal{V} = \{\mathcal{V}_l \cup \mathcal{V}_h\}$ is a finite set of nodes formed by *low-level nodes* $\mathcal{V}_l$ (i.e. instances of images or image sub-regions, having specific properties such as content - e.g. texture, shape, color, objects, etc... or more enhanced *features* and *general* or *domain-specific* metadata - e.g. author, title, description, tags, etc...) and *high-level nodes* $\mathcal{V}_h$ (i. e. instances of *general*, *domain-specific* or *image content* concepts); (ii) $\mathcal{E}$ is a subset of $(\mathcal{V} \times \mathcal{V})$; (iii) ρ is a function that associates to each couple of nodes a label indicating the kind of *relationship* between the related classes ρ_s, and its reliability degree $\rho_r \in [0, 1]$: $\rho : \mathcal{E} \rightarrow \langle \rho_s, \rho_r \rangle$.

Note that the effective use of this theory depends on the different kind of relationships that we can provide to the model, as described in the following. First, we provide a *similarity relationship*, that associates a couple of low-level nodes (images or sub-images) to a similarity degree, thus modeling the classical image matching algorithms based on low-level features (e.g. color, texture, shape, etc...) and/or semantic distances (e.g. taxonomy-based, Wu-Palmer, etc...) among metadata; a *membership relationship* verify if a given sub-image belongs to a given image; a *representativeness relationship* permits to associate those content features that better represent a given concept – using, for example, proper clustering or other classification algorithms that are able to determine what is the probability that an image is a valid representative of a concept; eventually, a *semantic relationship* may be applied to two high-level nodes (hypernym/hyponim, holonym/meronym, synonym, retrievable relations on lexical databases).

An example of a graph representing the extensional part of an image ontology related to some Italian landscapes is reported in Figure 1.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

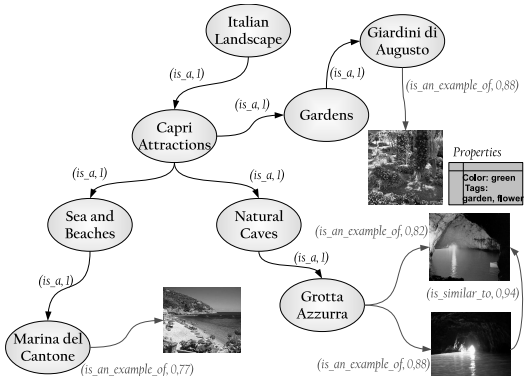


Fig. 1. An example of Image Ontology

2.2 The Building Process

As well noted in the literature, the critical part of an ontological framework is the effective construction of ontologies in a certain domain. Here, we describe a process useful to automatically build the graph representing our image ontology, by means of data-driven unsupervisioned approach. The proposed process, which is detailed in our previous works [9, 10], starts with the definition of an initial taxonomy containing a relevant concepts' instances hierarchy of the considered domain, that is represented by a subset of high level nodes. The definition is performed by experts in the domain of interest (domain-oriented approach).

After the taxonomy definition, we first extract images and the related textual description from publicly available image repositories, such as FLICKR. The images are then analyzed in order to obtain a low-level description in terms of content features, using classical Computer Vision techniques; the textual part is at the same time processed in order to discover textual *labels* that better reflect image semantics using NLP techniques and topic detection algorithms on the textual annotations coming from the considered image repositories. In the image analysis task, we use a *salient points* technique - based on the *Animate Vision* paradigm [5] - that exploits color, texture and shape information associated with those regions of the image that are relevant to human attention, in order to obtain a compact characterization (*Information Path*) that could be also used to evaluate the similarity between images, and for indexing issues. Eventually, apposite super-visioned classification algorithms are exploited to determine content features [5].

We then analyze the textual information [9] that is usually associated to an image. To these purposes, information coming from *tags* are analyzed in combination with *titles* and *descriptions* by suitable NLP techniques that overcome the linguistic and semantic heterogeneity of such information, in order to extract a set of *relevant keywords* which more effectively represent image content. In particular, the semantic processing can be decomposed into a set of sequential tasks: (i) the *Meta-Noise Filtering* - aims at rejecting inaccurate or irrelevant words in the input text; (ii) the *Named Entity Filtering* - is used to find particular words (*named entities*) in the input text and associate to

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

them the related semantic classes (person, organization, location, date, etc...); (iii) the *Linguistic Normalization* - performs the classical *stemming*; (iv) the *Linguistic Filtering* executes a *part of speech tagging* on the input text and selects only the words of *noun type*; (v) the *Word Sense Disambiguation (WSD)* - tries to automatically determine for each word the most suitable meaning by linking each word with the related WordNet synset; (vi) the *Topic Extraction* - aims at extracting a set of relevant keywords or *labels* with a *confidence* value considering both the *semantic similarity* among words, and the related *frequency* and *co-occurrence* of the words in the text.

Finally, the obtained knowledge in terms of images, low-level characteristics and labels is then merged into a graph that represents our image ontology. The proposed strategy is discussed in the following (see [10] for more details). Initially, all the images and the regions - whose relevant labels are associated to high-level nodes with a high grade of confidence - and that correspond to the leaves of the domain taxonomy - will be represented by apposite low-level nodes in the graph. In addition, couples of image nodes, whose similarity (computed by the *Information Path Matching* algorithm [5] and Wu/Palmer metric) is greater than a given threshold, will be connected by an edge having as reliability degree the related similarity measure. All the images belonging to the same concept are then clustered into different groups, which contain images that are more similar among themselves. We used as clustering procedure the *BEM* algorithm [5], that is recursively invoked to dynamically determine more fitting clusters without knowing a-priori the number of clusters themselves . Then we selected for each cluster the representative image as the closest one to all the other images of the cluster, and a suitable *representation probability* is associated to each representative image on the base of minimum and average distances. The process is iterated for each taxonomy leaf concept and the ontology is incrementally built: images belonging to different topics could be linked on the base of their similarity values allowing to *merge* the multimedia knowledge in a unique graph. Eventually, by a *Learning Tag Relevance algorithm* [6] (*Okapi BM25* ranking function), the topics that are more relevant – w.r.t. the content of images belonging to the same cluster (*winner topics*) – are *promoted* to be high-level nodes of the image ontology. The winner topics, whose relevance is greater than a threshold, are finally inserted as high-level nodes in the ontology and *linked*, from one hand to the image node that corresponds to the cluster centroid and, from the other one, to those nodes whose semantic distance (i.e. Wu/Palmer) is the minimum with respect to the current topic. If it is possible, the new ontology edge is labeled with the type of semantic relationship. In the case of topics that cannot be retrieved in WordNet, they are linked to the deepest high-level nodes of ontology from which they descend.

3 The System Architecture

The system architecture that supports the ontology building process is shown in figure 2. The user generates by a GUI an *OWL* file coding the initial taxonomy containing relevant concepts of the considered domain. Such a file is then the input of the *Information Fetcher* module that downloads images and the related annotations from the *Multimedia Repository*, using as *search keywords* the concepts related to the leaf nodes of the taxonomy and some filters on users.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A *Storage Engine* module receives such information and stores image annotations (title, author, tags, labels, etc...) in a dedicated *RDF Database* and it stores both raw data and low-level characteristics in a *Image Database*. Finally, the *Semantic Processor* and *Cluster Manager* analyze high and low level information in order to generate/update, by means of the *Ontology Manager* and in according to the described process, a graph which represents the final multimedia ontology (stored in a OWL format in a dedicated repository). As for implementation details, we notice that: (i) the initial taxonomy is generated by a JAVA desktop application that uses *Protege'* API; (ii) FLICKR has been chosen as the multimedia repository; (iii) the Information Fetching module has been implemented as a JAVA application that exploits the FLICKR API; (iv) the RDF and Image Database have been realized by *Sesame* and *PostegreSQL* DBMS, respectively; (v) the Indexing packages have been implemented by apposite JAVA packages that exploit Stanford NLP and Animate Vision libraries.

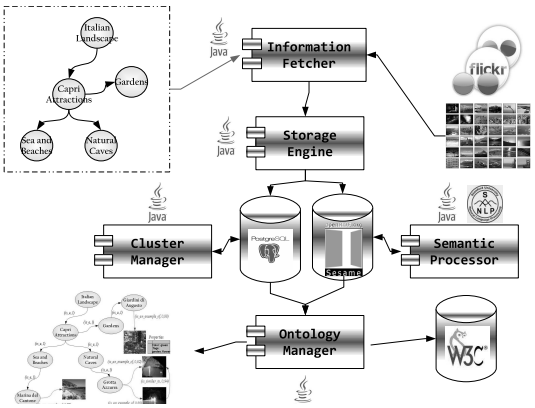


Fig. 2. System Architecture

4 A Case Study and Preliminary Experimental Results

Evaluating the "quality" of an ontology is an important issue both for ontology delevepers and for final users: this task allows to compare different ontologies describing the same domain in order to choose the more suitable one for a given application. For this aim, we have built an ontology related to *Capri*, a wonderful Italian island of the Sorrentine Peninsula, on the south side of the Gulf of Naples. A set of experts of natural and cultural attractions of Capri provided as initial taxonomy a graph containing the most relevant concepts in terms of high level nodes for the considered domain. The FLICKR repository has been queried using as search keywords the *logical AND* among concepts reported in the leaf nodes of the taxonomy and the one corresponding to the root node and exploiting some filters on user *id*s, in order to retrieve images really belonging to the domain.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Each retrieved image with the related annotations undergoes the described content-based analysis and semantic processing to determine the low-level description and the relevant labels to propagate in the ontology. Our efforts have been then devoted to produce experimental results in order to evaluate the *effectiveness* of the produced ontologies with respect to some generated by human domain experts w.r.t. different criteria: *Class Match Measure* (*CMM*), *DEnsity measure* (*DEM*), *Semantic Similarity Measure* (*SMM*), *BEtweenness Measure* (*BEM*) [1]. To compute the different metrics, we ask five persons [1] to describe in an exhaustive way and by means of an image ontology (concepts and photos selected from FLICKR) the main natural and cultural attractions of Capri, classifying them on the base of the related kind (sea and beaches, natural caves and gardens, squares and ancient villas).

Table 1. Values of the quality metrics for the *Capri* ontologies

Ontology	CMM	DEM	SSM	BEM	Avg Score
MOWIS	1	0,89	0,64	0,578	0,777
User A	0,68	0,89	1	1	0,892
User B	0,76	1	0,87	0,59	0,805
User C	0,8	0,89	0,846	0,578	0,78
User D	0,7	0,89	0,87	0,52	0,745
User E	0,63	0,94	1	0,315	0,72

Then, we compared such ontologies with that one produced by our system using the knowledge (images, tags, description and titles) associated to the same photos (about 1000) from FLICKR. For what the CMM metric computation concerns, we asked other 5 different user to provide a set of 50 common search terms for interesting Capri attractions (e.g. faraglioni, piazzetta, Jovis, marina grande, augusto, cave, azzurra, beach, anacapri) among FLICKR tags. Table 1 summarizes the experimental results for the different metrics obtained comparing the Capri ontology automatically generated by our system with respect to ontologies produced by the five human domain experts. The most expert users are indicated with A, B and C, while D and E represent the least experts. As we can see from the table, our ontology has a quality index very close to that of an ontology generated by humans quite experts on the considered domain. Finally, we measured the times (download, IP computation and clustering, semantic processing and tag propagation, ontology population) for building an image ontology depending on the number of fetched images [2]. We observed that The *Capri* Ontology, being composed by 1000 images, requires less than 10 minutes for its complete building (about two minutes if we do not consider the download from FLICKR), thus our apporach ensures a quite good scalability.

[1] In particular, we choose three persons more expert and other two less expert on the Capri attractions

[2] we use a Linux Ubuntu platform running on a 8GB RAM single CPU

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

5 Conclusions

In this paper, first we addressed the problem of building a multimedia ontology in an automatic way using annotated image repositories. Future work will be devoted to enlarge our experimentation to more significant case studies discussing the ontology maintenance problem and to make compatible output of the proposed framework with the latest languages for describing multimedia ontology (e.g. *M-OWL*).

6 Acknowledgments

The prototype realization has been carried out partially under the financial support of the FARO Programme in the framework of the *LINCE* Project: un sistema di Localizzazione/georeferenziazione degli INCidEnti stradali a basso costo.

References

1. Alani, H., Brewster , C.: Metrics for Ranking Ontologies. 4th Int. EON Workshop in 15th Int. World Wide Web Conf. (2006)
2. Benitez, A.B., Chang, S.F: Perceptual Knowledge Construction From Annotated Image Collections, International Conference on Multimedia & Expo. pp. 189-192 (2002)
3. Bertini, M., et al.: MOM: multimedia ontology manager. A framework for automatic annotation and semantic retrieval of video sequences, ACM International Conference on Multimedia, ACM Multimedia, pp. 787-788 (2006)
4. Bloehdorn, S., et al.: Knowledge representation for semantic multimedia content analysis and reasoning, Workshop on the Integration of Knowledge, Semantics and Digital Media technology (2004)
5. Boccignone, G., Chianese, A., Moscato, V., Picariello, A.: Context-sensitive Queries for Image Retrieval in Digital Libraries. Journal of Intelligent Information Systems, vol. 31, n. 1, pp. 53-84 (2008)
6. Golder S.A., Hubemann, A.: Usage Patterns of collaborative tagging systems. Information Science (2006)
7. Mishra, S., Ghosh, H.: Effective Visualization and Navigation in a Multimedia Document Collection using Ontology. Pattern Recognition and Machine Intelligence, pp. 501-506 (2009)
8. Kennedy, L., Naaman, M., Ahern, S., Nair, R., Rattenbury, T.: How flickr helps us make sense of the world: context and content in community-contributed media collections. ACM Multimedia, vol. 99, n.7, pp. 631-640 (2007)
9. Moscato, V., Penta, A., Persia, F., Picariello, A.: A System for Automatic Image Categorization. International Conference on Semantic Computing, pp. 624-629 (2009)
10. Moscato, V., Penta, A., Persia, F., Picariello, A.: A System for Building Multimedia Ontologies from Web Information Sources. IIR, pp. 89-93 (2010)
11. Paliouras, G., et al.: Bootstrapping Ontology Evolution with Multimedia Information Extraction, Knowledge-Driven Multimedia Information Extraction and Ontology Evolution. LNCS 6050, pp. 1-17 (2011)
12. Petridis, K., et al.: M-OntoMat-Annotizer: Image Annotation Linking Ontologies and Multimedia Low-Level Features. Knowledge-Based Intelligent Information and Engineering Systems (LNCS), vol. 4253, pp. 633-640 (2006)
13. Stamou, G. et al.: Multimedia annotations on the semantic Web- IEEE Multimedia, vol. 13, pp. 86-90 (2006)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Integration and Provenance of Cereals Genotypic and Phenotypic Data*,**

Domenico Beneventano, Sonia Bergamaschi, and Abdul Rahman Dannaoui

Department of Computer Science
University of Modena and Reggio Emilia
via Vignolese 905, 41125 Modena, Italy
firstname.lastname@unimore.it

Abstract. This paper presents the ongoing research on the design and development of a Provenance Management component, PM_{MOMIS}, for the MOMIS Data Integration System. PM_{MOMIS} aims to provide the provenance management techniques supported by two of the most relevant data provenance systems, the *Perm* and *Trio* systems, and extends them by including the data fusion and conflict resolution techniques provided by MOMIS.
PM_{MOMIS} functionalities have been studied and partially developed in the domain of genotypic and phenotypic cereal-data management within the CEREALAB project. The CEREALAB Data Integration Application integrates data coming from different databases with MOMIS, with the aim of creating a powerful tool for plant breeders and geneticists. Users of CEREALAB played a major role in the emergence of real needs of provenance management in their domain.

1 Introduction

The CEREALAB Data Integration Application has been developed to create a powerful tool for plant breeders and geneticists [21]. It stores genotypic and phenotypic cereal-data collected within the CEREALAB project and integrates them with already existing well known public data sources. The integrated database can help breeders and geneticists in: unravelling the genetics of economically important phenotypic traits, identifying and choosing molecular markers associated to key traits, and in choosing the desired parentals for breeding programs. The CEREALAB Data Integration Application development was one of the objectives of the CEREALAB and SITEIA projects and of the BIOGEST-SITEIA laboratory (www.biogest-siteia.unimore.it), funded by Emilia-Romagna (Italy) regional government, and aims to increase the competitiveness of Regional seed companies through the use of modern selection technologies, i.e. the Marker-Assisted Selection (MAS).

Data integration is obtained by using the MOMIS system (Mediator envirOnment for Multiple Information Sources), a framework to perform integration of structured

* Extended abstract of the paper "D. Beneventano, S. Bergamaschi, A.R. Dannaoui, N. Pecchioni: Integration and Provenance of Cereals Genotypic and Phenotypic Data, to appear as a Poster at Data Integration in the Life Sciences (DILS 2012)".

** This work is partially supported by the BIOGEST-SITEIA laboratory (www.biogest-siteia.unimore.it), funded by Emilia-Romagna (Italy) regional government.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

and semi-structured data sources [3,7]. MOMIS is characterized by a classical wrapper/mediator architecture: the local data sources contain the real data, while a Global Schema (GS) provides a *reconciled, integrated, read-only view* of the underlying sources. The GS and the mappings between the GS and the local sources are semi-automatically defined at design time by the Integration Designer component of the system [3]. After GS creation end-users can pose queries over this GS in a transparent way w.r.t. the local sources. MOMIS has been developed by the DBGROUP of the University of Modena and Reggio Emilia[1]. An open source version of the MOMIS system is delivered and maintained by the academic spin-off DataRiver[2].

As discussed in [21] the global classes of the *CEREALAB GS* plays the role of performing *data fusion*, i.e. the process of fusing multiple records representing the same real-world object into a consistent representation. Data fusion may involve the resolution of possible conflicts between data coming from different sources; several high level strategies to handle inconsistent data have been described and classified in [9]. MOMIS supports: *conflict avoiding* strategies (such as the *trust your friends* strategy which takes the value of a preferred source), *conflict ignoring* strategies (such as the *pass it on* strategy, which presents all values deferring conflict resolution to the user) and *resolution* strategies (such as the *meet in the middle* strategy which takes an average value). These strategies are implemented by means of *Resolution functions* in the *full outerjoin-merge operator* proposed in [23] and adapted to the MOMIS System in [7].

A requirement emerging from CEREALAB users, the breeders, was that in many cases they would prefer to give a look at the data coming from the local sources, i.e. they need *provenance*. A need to support detailed data provenance is one of the main requirements of biological data management identified in [19][3].

Lineage, or *provenance*, in its most general definition, describes where data came from, how it was derived and how it was modified over time. Lineage provides valuable information that can be exploited for many purposes, ranging from simple statistical resumes presented to the end-user to more complex applications, such as, managing data uncertainty or identifying and correcting data errors. Lineage has been studied extensively in data warehouse systems [11], but it is still an open research problem in Data Integration systems [15,14].

In [4] we introduced the notion of provenance into MOMIS, by defining the provenance for the *full outerjoin-merge operator*; this definition is based on the concept of *PI-CS*-provenance (Perm Influence Contribution Semantics) proposed in *Perm* (Provenance Extension of the Relational Model) [12] to produce more precise provenance information for outerjoins. Another important reason behind the choice of using the *PI-CS*-provenance, was that it is fully implemented in an open-source provenance management system that is capable of computing, storing and querying provenance for relational databases. At present, we are using the *Perm* system as the SQL engine of MOMIS, so that to obtain the provenance in our CEREALAB Application.

[1] `http://www.dbgroup.unimore.it`

[2] `http://www.datariver.it`

[3] This article is an extract from the Report of the NSF Workshop on Data Management for Molecular and Cell Biology, edited by H. V. Jagadish and Frank Olken he workshop was held at the National Library of Medicine, Bethesda, MD, Feb. 2-3,2003)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2 The MOMIS Data Integration System

A MOMIS Data Integration System [3,7,8] is constituted by: a set of *local schemas* $\{LS_1, \ldots, LS_k\}$, a *global schema* GS and *Global-As-View* (*GAV*) mapping assertions [20] between GS and $\{LS_1, \ldots, LS_k\}$. For each global class G of GS, a *Mapping Table* (*MT*) is defined, whose columns represent the set of local classes $\{L_1, \ldots, L_n\}$ of G; an element $MT[GA][L]$ represents the local attribute of L which is mapped onto the global attribute GA, or $MT[GA][L]$ is empty (there is no local attribute of L mapped onto the global attribute GA)[4]. *GAV* mapping assertions are expressed by specifying for each G a *mapping query* over its local classes, which defines the instance of G.

2.1 The CEREALAB Data Integration Application

In [21] we described the MOMIS semi-automatic approach to build the *GS* of the CEREALAB Data Integration Application. In this paper, we focus on the *GermPlasm* global class (see Figure 1) obtained by integrating two local classes `A` and `B` which store data about germplasms coming from two different data sources: `A` stores data obtained by experimental results within the CEREALAB project, `B` stores data coming from other public data sources (see [21] for a fully description of these sources).

The attributes of these tables represent the following information: `GPN`: the name of a variety; `FHB`: the resistance of the germplasm to the *FHB* disease (Fusarium Head Blight) (as values are S=susceptible, MR= Moderately Resistant and R= Resistant); *Type*: the variety's type; `yield` : the *grain yield* expressed in tons/hectare.

Figure 1 shows the Mapping Table of the global class `GERMPLASM`, with schema `GERMPLASM(GPN,Yield,FHB,Type)`, obtained by integrating `A` and `B`. As discussed in [21] `GERMPLASM` plays the role of performing *data fusion*. To identify multiple local tuples coming from local classes and representing the same real-world object, we assume that error-free and shared object identifiers exist among different sources: two local tuples with the same object identifier ID indicate the same object in different sources; thus we can use L^{id} to denote the tuple t of a local class L with ID equal to id, i.e. $t[ID] = id$. In our example, we assume `GPN` as an object identifier; then the first tuple of the local class `A`, i.e. the tuple with `GPN` $= Eureka$, will be denoted with $\texttt{A}^{Eureka}$. For conflicting attributes the following strategies are used:

- `FHB` (*trust your friends* strategy): `FHB`=COALESCE(`A.FHB`,`B.FHB`).
  Conflicts are solved preferring the data coming from the `A` Italian source to offer the Italian breeders information from Italian studies and so nearest to their needs;
- `Type` (*pass it on* strategy): `Type`= ALLVALUES(`A.type`, `B.type`)
  A germplasm type is unique, so when we find two different types for the same

[4] For the sake of simplicity, we consider a simplified version of the MOMIS framework proposed in [3,7], where $MT[A][L]$ is a set of local attributes and *Data Transformation Functions* specify how local attribute values have to be transformed into corresponding global attribute values. Moreover we assume $S(G) = \cup_i S(L_i)$, i.e. global and local attribute names are the same. Finally, both the global and the local schemas are expressed in the ODL_{I^3} language [8]. However we consider both GS and LS_i as relational schemas, but we will refer to their elements respectively as global and local classes to comply with the MOMIS terminology.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Local classes

A (GermPlasmA)

GPN	yield	FHB	type
Eureka	18	MR	
Fortuna	7	MR	
Mentana		S	line
Kenora	20	MR	landrace
Oasis	21	MR	cultivar

B (GermPlasmB)

GPN	yield	FHB	type
Eureka	6	S	cultivar
Fortuna	15	S	landrace
Mentana	20	MR	line
Kenora			cultivar

Mapping Table of the global class GERMPLASM

GERMPLASM	A	B
GPN	GPN	GPN
Yield	yield	yield
FHB	FHB	FHB
Type	type	type

Fig. 1. Example: two local classes with two conflicting attributes

Mapping Query of GERMPLASM

```
SELECT GPN = GPN,
       Yield=AVG(A.yield, B.yield)
       FHB = COALESCE(A.FHB,B.FHB),
       Type = ALLVALUES(A.type, B.type)
FROM   A FULL OUTER JOIN B
       USING (GPN)
```

Instance of GERMPLASM

GPN	Yield	FHB	Type
Eureka	12	MR	cultivar
Fortuna	11	MR	landrace
Mentana	20	S	line
Kenora	20	MR	landrace,cultivar
Oasis	21	MR	cultivar

Fig. 2. Mapping Query and Instance of the global class GERMPLASM

germplasm we know that uncertain data are present. In the case of *Kenora*: we have two types but we do not know which value is correct. To avoid choosing the wrong value all values are maintained and conflict resolution is deferred to the user;

– Yield (*meet in the middle* strategy): Yield=AVG(A.yield, B.yield) represents the average of grain yield in the local classes.

Finally, the mapping query of the global class GERMPLASM (and then the instance of GERMPLASM) is defined by means of the *full outerjoin-merge operator*. Intuitively, as shown in Figure 1, this corresponds to the following two operations: (1) Computation of the *full outerjoin*, on the basis of the shared object identifier, of the *local classes* of GERMPLASM; (2) Application of the *Resolution Functions* [5].

2.2 Provenance at present implemented in the MOMIS system

In [4] the concept of *PI-CS*-Provenance was applied and extended to the full outerjoin-merge operator. As an example, let us consider a query on GERMPLASM (the user is searching for the types of the varieties that are resistant to the fusarium head blight):

```
TYPE_MR = SELECT  DISTINCT Type
            FROM  GERMPLASM
            WHERE FHB = 'MR'
```

[5] COALESCE is the (standard SQL) function which returns its first non-null value; AVG is a (non standard SQL) function to compute the average value; ALLVALUES is a (non standard SQL) function which returns all non-null values.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

Type	*PI-CS* Provenance as a set of witness lists
landrace	$\{ \langle \mathtt{A}^{Fortuna}, \mathtt{B}^{Fortuna} \rangle \}$
cultivar	$\{ \langle \mathtt{A}^{Eureka}, \mathtt{B}^{Eureka} \rangle, \langle \mathtt{A}^{Oasis}, \perp \rangle \}$
landrace,cultivar	$\{ \langle \mathtt{A}^{Kenora}, \mathtt{B}^{Kenora} \rangle \}$

Relational Representation of the *PI-CS* Provenance

Type	A.GPN	A.yield	A.FHB	A.type	B.GPN	A.yield	B.FHB	B.type
landrace	Fortuna	7	MR		Fortuna	15	S	landrace
cultivar	Eureka	18	MR		Eureka	6	S	cultivar
cultivar	Oasis	21	MR	cultivar				
landrace,cultivar	Kenora	20	MR	landrace	Kenora			cultivar

Fig. 3. Example: *PI-CS* Provenance for the query TYPE_MR

The *PI-CS*-Provenance of an output tuple is a set of *witness lists*, where each witness list represents one combination of local tuples that were used together to derive the output tuple; a witness list contains a local tuple from each local class or the special value $\perp$, indicating that no tuple from a local class was used to derive the output tuple (useful in modeling outerjoins). For example, the *PI-CS*-Provenance of the output tuple cultivar (see Figure 3) is a set of two *witness lists*, where the second one $\langle \mathtt{A}^{Oasis}, \perp \rangle$ indicates that $\mathtt{A}^{Oasis}$ paired with no tuples of B is a possible derivation of the output tuple. Witness lists are represented in a relational form, as shown in Figure 3: each witness list of an output tuple is represented by a single tuple.

The main drawback of this solution is that often conflicting values represent *alternative*. For example if we want to select other germplasms of the same type of *Kenora*, the two values landrace and cultivar must be considered as *alternative values*; if not, we might obtain Fortuna and Eureka as germplasms of the same type. Our proposal to overcome this drawback is to consider the output of the full outer join merge operator as an *uncertain relation* and then manage it with a system that supports uncertain data and data lineage, the *Trio* system [1,6].

3 Provenance based Conflict Handling Strategies

The *Trio* system is based on the *ULDB* data model [5], which extends the relational model with:

- *Alternatives*, representing uncertainty about the contents of a tuple. ULDB uncertain relations have a set of *certain* attributes and a set of *uncertain* attributes; each tuple in a ULDB relation has one value for each certain attribute, and a set of possible values for the uncertain attributes.
- *maybe* ('?') annotations, representing uncertainty about the presence of a tuple.
- *Lineage*, connecting tuple-alternatives to other tuple-alternatives from which they were derived: *Trio*-Provenance is a boolean formula λ over tuple-alternatives.
- *Confidences*: numerical confidence values optionally attached to alternatives.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

(A) Global class GERMPLASM

GPN	Yield	FHB	Type	
Eureka	12	MR	cultivar	?
Fortuna	11	MR	landrace	?
Mentana	20	S	line	?
Kenora	20	MR	landrace || cultivar	?
Oasis	21	MR	cultivar	

(B) query TYPE_MR

Type	
landrace	?
cultivar	
landrace || cultivar	?

Fig. 4. Global class GERMPLASM and query TYPE_MR as *uncertain relations*

Intuitively, these concepts might be applied to our data fusion context as follows. As shown in Figure 4.A, the global class GERMPLASM is considered as an *uncertain relation* where *non-conflicting* attributes and *solved-conflicting* attributes (like FHB and Yield) are modelled as *certain* attributes and *not-solved-conflicting* attributes (like Type) are modelled as *uncertain* attributes. The global tuple identified by GPN $=$ $Kenora$, denoted by $Kenora$, has two *alternatives*, $(Kenora, 1)$ and $(Kenora, 2)$. A global tuple coming from local tuples with conflicting values (solved or not solved) is annotated with '?' ; in the example, all global tuples are *maybe* tuples '?', with the exception for $Oasis$. The *Trio*-Provenance of a global tuple-alternative is a boolean formula over the local tuples from which the alternatives were derived; for example:
$\lambda(Kenora, 1) = \mathtt{A}^{Kenora}$: this alternative derives from a local tuple in A;
$\lambda(Kenora, 2) = \mathtt{A}^{Kenora} \wedge \mathtt{B}^{Kenora}$: this alternative derives from the conjunction of two local tuples. At present *Confidences* are not considered in our framework.

How can we implement the three above TRIO concepts in PM_{MOMIS}? Can we obtain the GERMPLASM *uncertain relation* shown in the example by means of the *Trio* system?

GERMPLASM computation is based on outerjoins and uses resolution functions, thus such computation is not simple as in *Trio* outerjoins are not allowed[6] and resolution functions have to be implemented in *Trio* (as we already did with a strong effort in *Perm*). For this reason, our choice is to use PERM for computing global classes and Trio for querying global classes. The computation of the full outer join merge operator is extended to obtain global classes including conflicts as *uncertain relations* with their related *TRIO*-Provenance; this computation is discussed in [2] and its result is intuitively shown in Figure 4.A. Thus, the *Trio* system[7] may be used to execute queries on global classes (i.e. exploiting the uncertain database theory); intuitively, the uncertain relation resulting from query TYPE_MR_S is shown in Figure 4.B. In this way, the user knows that only cultivar (the second tuple) is coming from non-conflicting local tuples, since it is not a *maybe* tuples; then he may obtain the provenance of this tuple either[8] at

[6] In a *TRIO* database U, *base tables* are uncertain relations and the result of a relational query Q on U is an uncertain relation: in [16] the problem of incorporating outerjoins into uncertain databases is considered but the authors argued that standard possible-worlds semantics may be inappropriate for outerjoins.

[7] The *Trio* source code is freely available and the system is based also on PostgreSQL.

[8] *TRIO*-Provenance is a multilevel (transitive) relationships: formulas specify direct derivations, but when the alternatives in a formula are themselves derived from other alternatives, it is possible to recursively expand a formula until it specifies local tuples only.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

the level of global classes, $(Eureka, 1) \vee (Oasis, 1)$, and at the level of local classes, $(\mathrm{A}^{Eureka} \wedge \mathrm{B}^{Eureka}) \vee \mathrm{A}^{Oasis}$.

A relevant application and extension of the *Trio* system to query conflicting data, is to use the possible instances of a query to provide the user with different *search strategies* for querying the Global Schema. As an example (the user is searching for the types of the varieties with a yield value greater than 11):

```
HIGH_PROD_TYPE =
   SELECT DISTINCT Type
   FROM GERMPLASM
   WHERE Yield > 11
```

`HIGH_PROD_TYPE` as an *uncertain* relation

Type	
line	?
cultivar	
landrace || cultivar	?

The user, by interacting with the framework, might obtain: At first, only *consistent global tuples* coming from non-conflicting local tuples are viewed : { cultivar }. Then, after the application of conflict handling strategies (`yield` solved with `AVG`, `Type` considered as an uncertain attribute) the uncertain relation `HIGH_PROD_TYPE`, provides two different answers: (1) tuples in *every* possible instance: { cultivar,line }; (2) tuples in *some* possible instance: { cultivar,line,landrace}.

4 Conclusion and Future Work

In this paper we presented the ongoing research on the design and development of a Provenance Management component, $\mathrm{PM}_{\mathrm{MOMIS}}$, for the MOMIS System. $\mathrm{PM}_{\mathrm{MOMIS}}$ functionalities have been studied and partially implemented in the domain of genotypic and phenotypic cereal-data management within the CEREALAB project.

Several notions of provenance for database queries have been proposed and studied in the past years, see [10] for a survey. Among these approaches, one of the most expressive ones is the *Provenance Semiring* [13]; an extension of the *Provenance Semiring* to schema mappings is used in ORCHESTRA data sharing system [18]; as in our framework, provenance in ORCHESTRA is also used to perform reconciliation based on user preferences on the sources the data come from; moreover, the ORCHESTRA system is being prototyped in applications with biological collaborators [17]. *Provenance Semirings* and ORCHESTRA are then important references for our future work.

Another important references for our future work is the Open Provenance Model (OPM) [22], which aims to define a generic and comprehensive representation of data provenance and which is becoming a standard to share provenance information. OPM represents provenance through a graph; a graphical representation of provenance graph may be used to formulate queries from a visual and intuitive interface, and then enabling end-users (plant scientists) to be able to directly query the provenance information.

An interesting idea of provenance application in the biology community is sketched in [19]: mechanisms similar to the bibliographic citation index for articles and authors are needed to acknowledge ”publication” of datasets in shared databases, so as to encourage rapid, effective sharing of data; data management support for tracking data provenance can provide the analog of citations. Following this idea we are ”ranking” the local sources integrated in the CEREALAB application on the basis of users’ queries.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

References

1. Agrawal, P., Benjelloun, O., Sarma, A.D., Hayworth, C., Nabar, S., Sugihara, T., Widom, J.: Trio: a system for data, uncertainty, and lineage. In: VLDB '06. pp. 1151–1154 (2006)
2. Beneventano, D.: Provenance based conflict handling strategies. In: Data Quality in Data Integration Systems, DASFAA-Workshop, 18 April, South Korea (2012), to appear
3. Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M.: Synthesizing an integrated ontology. IEEE Internet Computing 7(5), 42–51 (2003)
4. Beneventano, D., Dannoui, A.R., Sala, A.: Data lineage in the momis data fusion system. In: ICDE-Workshops, April 11-16, 2011, Hannover, Germany. pp. 53–58 (2011)
5. Benjelloun, O., Sarma, A.D., Halevy, A., Widom, J.: Uldbs: databases with uncertainty and lineage. In: VLDB '06. pp. 953–964. VLDB Endowment (2006)
6. Benjelloun, O., Sarma, A.D., Hayworth, C., Widom, J.: An introduction to uldbs and the trio system. IEEE Data Eng. Bull. 29(1), 5–16 (2006)
7. Bergamaschi, S., Beneventano, D., Guerra, F., Orsini, M.: Data integration. In: Handbook of Conceptual Modeling: Theory, Practice and Research Challenges. Springer Verlag (2011)
8. Bergamaschi, S., Castano, S., Vincini, M., Beneventano, D.: Semantic integration of heterogeneous information sources. Data Knowl. Eng. 36(3), 215–249 (2001)
9. Bleiholder, J., Naumann, F.: Data fusion. ACM Comput. Surv. 41(1), 1–41 (2008)
10. Cheney, J., Chiticariu, L., Tan, W.C.: Provenance in databases: Why, how, and where. Foundations and Trends in Databases 1(4), 379–474 (2009)
11. Cui, Y., Widom, J., Wiener, J.L.: Tracing the lineage of view data in a warehousing environment. ACM Trans. Database Syst. 25(2), 179–227 (2000)
12. Glavic, B., Alonso, G.: Perm: Processing provenance and data on the same data model through query rewriting. In: ICDE '09. pp. 174–185 (2009)
13. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 31–40. PODS '07, ACM, New York, NY, USA (2007)
14. Halevy, A., Li, C.: Information integration research: Summary of nsf idm workshop breakout session. NSF IDM Workshop (2003)
15. Halevy, A., Rajaraman, A., Ordille, J.: Data integration: the teenage years. In: VLDB '06. pp. 9–16. VLDB Endowment (2006)
16. Ikeda, R., Widom, J.: Outerjoins in uncertain databases. In: Management of Uncertain Data (MUD). Stanford InfoLab (2009), http://ilpubs.stanford.edu:8090/925/
17. Ives, Z.G.: Data integration and exchange for scientific collaboration. In: Proceedings of the 6th International Workshop on Data Integration in the Life Sciences. pp. 1–4. DILS '09, Springer-Verlag, Berlin, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-02879-3_1
18. Ives, Z.G., Green, T.J., Karvounarakis, G., Taylor, N.E., Tannen, V., Talukdar, P.P., Jacob, M., Pereira, F.: The orchestra collaborative data sharing system. SIGMOD Rec. 37(3), 26–32 (Sep 2008), http://doi.acm.org/10.1145/1462571.1462577
19. Jagadish, H.V., Olken, F.: Database management for life sciences research. SIGMOD Rec. 33, 15–20 (June 2004), http://doi.acm.org/10.1145/1024694.1024697
20. Lenzerini, M.: Data integration: A theoretical perspective. In: PODS. pp. 233–246 (2002)
21. Milc, J., Sala, A., Bergamaschi, S., Pecchioni, N.: A genotypic and phenotypic information source for marker-assisted selection of cereals: the cerealab database. Database (2011)
22. Moreau, L., Freire, J., Futrelle, J., Mcgrath, R.E., Myers, J., Paulson, P.: The open provenance model: An overview. In: Provenance and Annotation of Data and Processes, pp. 323–326. Springer-Verlag (2008)
23. Naumann, F., Freytag, J.C., Leser, U.: Completeness of integrated information sources. Inf. Syst. 29(7), 583–615 (2004)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A Short History of Schema Mapping Systems*

Giansalvatore Mecca[1], Paolo Papotti[2], and Donatello Santoro[1]

[1] Università della Basilicata – Potenza, Italy
[2] Qatar Computing Research Institute (QCRI) – Doha, Qatar

1 Introduction

There are many applications that need to exchange, correlate, and integrate heterogenous data sources. These information integration tasks have long been identified as important problems and unifying theoretical frameworks have been advocated by database researchers [5].

To solve these problems, a fundamental requirement is that of manipulating *mappings* among data sources. The application developer is typically given two schemas – one called the source schema, the other called the target schema – that can be based on different models, technologies, and rules. Mappings, also called *schema mappings*, are expressions that specify how an instance of the source repository should be translated into an instance of the target repository. In order to be useful in practical applications, they should have an executable implementation – for example, by means of SQL queries or XQuery scripts. This latter feature is a key requirement in order to embed the execution of the mappings in more complex application scenarios, that is, in order to make mappings a plug and play component of integration systems.

Traditionally, data transformation has been approached as a manual task requiring experts to understand the design of the schemas and write scripts to translate data. As this work is time-consuming and prone to human errors, mapping generation tools have been created to make the process more abstract and user-friendly, thus easier to handle for a larger class of people.

In this paper, we outline a history of the different phases that have characterized the research about automatic tools and techniques for schema mappings and data exchange. We identify three different ages, as follows.

The Heroic Age The heroic age of schema-mappings research started ten years ago with the seminal papers about the Clio system [17, 19]: a first generation of tools was proposed to support the process of generating complex logical dependencies – typically *tuple-generating dependencies* [4] – based on a user-friendly abstraction of the mapping provided by the users. Once the dependencies are computed, these tools transform them into executable scripts to generate a target solution in a scalable and portable way.

Early schema-mapping tools proved to be very effective in easing the burden of manually specifying complex transformations, and were successfully transferred, to some extent, into commercial products (e.g., [13]). However, several

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

years after the development of the initial Clio algorithm, researchers realized that a more solid theoretical foundation was needed in order to consolidate practical results obtained on schema mappings systems. This consideration has motivated a rich body of research about *data exchange* that characterizes the next age.

The Silver Age Data exchange [5, 8, 19] formally studies the semantics of generating an instance of a target database given a source database and a set of mappings. It has formalized the notion of a *data exchange problem* [8], and has established a number of results about its properties.

After the first data exchange studies, it was clear that a key problem in schema-mappings tools was that of the *quality of the solutions*. In fact, there are many possible solutions to a data-exchange problem, and these may largely differ in terms of size and contents. The notion of the *core of the universal solutions* [10] was identified as the "optimal" solution, since it is the smallest among the solutions that preserve the semantics of the mapping.

In the last three years an intermediate generation of tools [16, 21] have emerged to address the problem of generating solutions of optimal quality, while guaranteeing at the same time the portability and scalability of the executable scripts. Nevertheless, despite the solid results both in system and theory fields, the adoption of mapping systems in real-life integration applications, such as ETL workflows or Enterprise Information Integration, has been quite slow. This seems to be due to three main factors: (*a*) these systems were not able, at first, to handle functional dependencies over the target, which is a key requirement in order to obtain solutions of quality; (*b*) the results were obtained primarily for relational databases, and did not extend to nested models and XML; (*c*) finally, there was no open-source schema-mapping tool available to the community.

The Golden Age A number of recent results [14, 7], along with the public availability of the first open-source mapping tools – like ++Spicy [15] and OpenII [20] – seem to be a promising starting point towards the solution of these problems and the beginning of a new, golden age for mapping tools. These works, along with others [1, 11], are giving new vitality to schema-mappings research and suggest new applications, beyond traditional data exchange and data integration tasks.

In the following, we first describe the three ages in Section 2 and we then draw some conclusions in Section 3.

2 A History of Schema Mappings In Three Ages

2.1 The Heroic Age

The first mapping generation tools were created to make the process of defining transformations among schemas easier and more effective with respect to manually developed scripts. This first generation of tools includes primarily Clio [12, 13, 17, 19] and systems [6, 20] which incorporate a Clio-like first-generation mapping module. We summarize the features of these early mapping tools as follows.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Value Correspondences The goal of simplifying the mapping specification was pursued by introducing a GUI that allows users to draw arrows, or *correspondences*, between schemas in order to define the desired transformation.

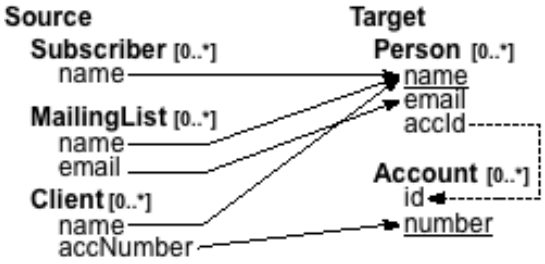


Fig. 1. Schema mapping scenario.

Consider the example shown in Figure 1, where data from multiple sources should be transformed into data for a target schema with a foreign key constraints between two relations. A correspondence maps atomic elements of the source schema to elements of the target schema, independently of the underlying data model or of logical design choices, and can be derived automatically with schema matching components. Notice that, while correspondences are easy to create and understand, they are a "poor" language to express the full semantics of data transformations. For this reason, a schema mapping tool should be able to interpret the semantics the user wants to express with a set of correspondences.
Mapping Generation Based on value correspondences, mapping systems generate logical dependencies to specify the mapping. These dependencies are logical formulas of two forms: *tuple-generating dependencies* (tgds) or *equality-generating dependencies* (egds). There are two classes of constraints. *Source-to-target tgds* (s-t tgds), i.e., tgds that use source relations in the premise and target relations in the conclusion, are used to specify which tuples should be present in the target based on the tuples that appear in the source. In an operational interpretation, they state how to "translate" data from the source to the target. Target schemas are also modeled with constraints: *target tgds*, i.e., tgds that only use target symbols, are used to specify foreign-key constraints on the target; while *target egds* are used to encode functional dependencies, such as keys, on the target database.

The mapping scenario in Figure 1 has three different source tables: (*i*) a table about subscribers of a service; (*ii*) a table with the email addresses of the people receiving the company mailing list; (*iii*) a table about clients and their check accounts. The target schema contains two tables, one about persons, the second about accounts. On these tables, we have two keys: *name* is a key for the persons, while *number* is a key for the accounts. Based on the correspondences drawn in Figure 1, a Clio-like system would generate the following set of tgds:

Source-to-Target Tgds
$m_1.\ \forall n\colon \mathit{Subscriber}(n) \rightarrow \exists Y_1, Y_2 : \mathit{Person}(n, Y_1, Y_2)$
$m_2.\ \forall n, e\colon \mathit{MailingList}(n, e) \rightarrow \exists Y_1 : \mathit{Person}(n, e, Y_1)$
$m_3.\ \forall n, acc\colon \mathit{Client}(n, acc) \rightarrow \exists Y_1, Z\colon (\mathit{Person}(n, Y_1, Z) \wedge \mathit{Account}(Z, acc))$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

In addition, the following egds translate the key constraints on the target schema:

TARGET EGDS
e_1. $\forall n, e, a, e', a' : Person(n, e, a) \wedge Person(n, e', a') \rightarrow (e = e') \wedge (a = a')$
e_2. $\forall n, i, i' : Account(i, n) \wedge Account(i', n) \rightarrow (i = i')$

Mapping Execution via Scripts To execute the mappings, schema-mapping systems rely on the traditional *chase procedure* [8]. The chase is a fixpoint algorithm which tests and enforces implication of data dependencies, such as tgds, in a database. To be more specific, a first-generation system, after the mappings had been generated, would discard the target dependencies, and translate the source-to-target ones under the form of an SQL or XQuery script that implements the chase and can be applied to a source instance to return a solution.

Notice, in fact, that the chase of a set of s-t tgds on I can be naturally implemented using SQL. Given a tgd $\phi(\overline{x}) \rightarrow \exists \overline{y}(\psi(\overline{x}, \overline{y}))$, in order to chase it over I we may see $\phi(\overline{x})$ as a first-order query Q_ϕ with free variables $\overline{x}$ over the source database. We execute $Q_\phi(I)$ using SQL in order to find all vectors of constants that satisfy the premise and we then insert the appropriate tuple into the target instance to satisfy $\psi(\overline{x}, \overline{y})$. Skolem functions [19] are typically used to automatically "generate" some fresh nulls for $\overline{y}$.

However, these systems suffer from a major drawback: they did not have a clear theoretical foundation, and therefore it was not possible to reason about the quality of the solutions.

2.2 The Silver Age

Data exchange was conceived as an attempt to formalize the semantics of schema mappings. It formalized many aspects of the mapping execution process, as follows.

Data Exchange Fundamentals. In a data-exchange setting, the source and target databases are modeled by having two disjoint and infinite sets of values that populate instances: a set of *constants*, CONST, and a set of *labeled nulls*, NULLS [8]. Labeled nulls are used to "invent" values according to existential variables in tgd conclusions. The reference data model is the relational one.

A *mapping scenario* (also called a *data exchange scenario* or a *schema mapping*) is a quadruple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$, where $\mathbf{S}$ is a source schema, $\mathbf{T}$ is a target schema, Σ_{st} is a set of source-to-target tgds, and Σ_t is a set of target dependencies that may contain tgds and egds [8].

Given two disjoint schemas, $\mathbf{S}$ and $\mathbf{T}$, we denote by the pair $\langle \mathbf{S}, \mathbf{T} \rangle$ the schema $\{S_1 \ldots S_n, T_1 \ldots T_m\}$. If I is an instance of $\mathbf{S}$ and J is an instance of $\mathbf{T}$, then the pair $\langle I, J \rangle$ is an instance of $\langle \mathbf{S}, \mathbf{T} \rangle$.A target instance J is a *solution* [8] of $\mathcal{M}$ and a source instance I iff $\langle I, J \rangle \models \Sigma_{st} \cup \Sigma_t$, i.e., I and J together satisfy the dependencies. Given a mapping scenario $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$, a *pre-solution* for $\mathcal{M}$ and a source instance I is a solution over I for scenario $\mathcal{M}_{st} = (\mathbf{S}, \mathbf{T}, \Sigma_{st})$, obtained from $\mathcal{M}$ by removing target constraints. In essence, a pre-solution is a solution for the s-t tgds only, and it does not necessarily enforce the target

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

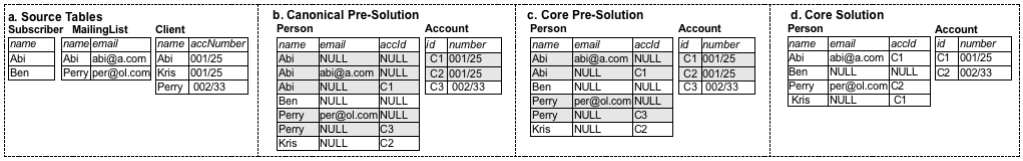


Fig. 2. Source instance (a) and three possible solutions (b-d).

constraints. Given the source data in Figure 2.a, the canonical pre-solution is reported in Figure 2.b. A mapping scenario may have multiple solutions on a given source instance: each tgd only states an inclusion constraint and does not fully determine the content of the target. Among the possible solutions we restrict our attention to *universal* solutions, which only contain information from I and $\Sigma_{st} \cup \Sigma_t$. Universal solutions have a crucial property: they have a *homomorphism* (i.e., a constant-preserving mapping of values) into all the solutions for a data exchange problem. Intuitively, this guarantees that the solution does not contain any arbitrary information that does not follow from the source instance and the mappings. Under a condition of weak acyclicity of the target tgds, an universal solution for a mapping scenario and a source instance can be computed in polynomial time by resorting to the classical *chase procedure* [8]. A solution generated by the chase is called a *canonical solution*. In light of this, we may say that early mapping systems were restricted to generate *canonical pre-solutions*, since they chased s-t tgds only.

Tools of the Intermediate Generation. Once the theory of data-exchange had become mature, it was clear that producing solutions of quality was a critical requirement. The notion of a *core solution* [10] was formalized as the "optimal" solution, since it is universal, and among the universal solutions is the one of the smallest size. In our example, the core solution is reported in Figure 2.d.

Sophisticated algorithms were developed to post-process a canonical solution generated by a schema-mapping tool, and minimize it to find its core [10, 18]. These tools have the merit of being very general, but fail to be scalable: even though the algorithms are polynomial, their implementation requires to couple complex recursive computations with SQL to access the database, and therefore do not scale nicely scale to large databases. In fact, empirical results show that they are hardly usable in practice due to unacceptable execution times for medium size databases [16].

It was therefore clear that, in order to preserve the effectiveness and generality of mapping tools, reasoning about the mapping was necessary. First, a number of approaches were proposed to optimize schema mappings in order to improve the efficiency of their execution and manipulation. In fact, schema mappings may present redundancy in their expressions, due for example to the presence of unnecessary atoms or unrelated variables, thus negatively affecting the data management process [9, 18].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A different approach to the generation of core solutions was undertaken in [16, 21]. In these proposals, scalability is a primary concern. Given a mapping scenario composed of source-to-target tgds (s-t tgds), the goal is to rewrite the tgds in order to generate a runtime script, for example in SQL, that, on input instances, materializes core solutions. This is a key requirement in order to embed the execution of the mappings in more complex application scenarios, that is, in order to make data-exchange techniques a real "plug and play" feature of integration applications. +Spicy [16] is an example of mapping tool of this generation. These works exploit the use of *negation* in the premise of the s-t tgds to rewrite them intercepting possible redundancy. Consider again our running example; algorithms for SQL core-generation would rewrite m_1 to make sure that no redundant data are copied to the target from the relation *Subscriber*:

$$m_1'.\ \mathit{Subscriber}(n) \wedge \neg(\mathit{MailingList}(n, E)) \wedge \neg(\mathit{Client}(n, A)) \rightarrow \mathit{Person}(n, Y_1, Y_2)$$

Experiments [16] show that, in the computation of the core solution, with executable scripts there is a gain in efficiency of orders of magnitude with respect to the post-processing algorithms. This is not surprising, as these mapping rewriting approaches preserve the possibility to execute transformations in standard SQL, with the guarantee of scalability to large databases and of portability to existing applications.

However, these tools still have some serious limitations, that prevent their adoption in real-life scenarios. We may summarize these limitations as follows.

(*a*) *They have limited support for target constraints.* Handling target constraints – i.e., keys and foreign keys, represented by *egds* and *target tgds* [8], respectively – is a crucial requirement in many mapping applications. Notice that foreign-key constraints were at the core of the original schema-mapping algorithms, and, under appropriate hypothesis, can always be rewritten as part of the source-to-target tgds [9]. Unfortunately this is not the case for target edgs.

Consider again the running example; the best a tool from this generation can obtain with executable scripts is the core pre-solution reported in Figure 2.c, where the redundancy coming from the source-to-target tgds has been removed, but the solution lacks the enforcement of the target key constraints.

(*b*) *They are limited to relational scenarios, and cannot handle XML or nested datasets.* This is a consequence of the fact that data-exchange research has primarily concentrated on the relational setting, and for a long time no notion of data exchange for more complex models was available. In a way, this is a setback with respect to the early systems, which had supported nested relations since the beginning with a pragmatical approach. In fact, they were able to produce results for XML setting, but without the precise definition of quality that core solutions provide. It is interesting to note that a benchmark for mapping systems has been recently proposed [2]. However, none of the tools of the intermediate generation can be evaluated using the benchmark – for example in order to compare the quality of their solutions – since most of the scenarios in the benchmark refer to nested structures, and these systems are not capable to generate core solutions for a nested data model.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2.3 Time for a Golden Age

Recent results have faced these problems and paved the way towards the emergence of a new-generation of schema-mapping and data-exchange tools.

A first important advancement is related to the management of functional dependencies over the target. Although it is not always possible, in general, to enforce a set of egds using a first-order language as SQL, it has been proposed a best-effort algorithm that rewrites the above mapping into a new set of s-t tgds that directly generate the target tuples that are produced by chasing the original tgds first and then the egds [14]. As egds merge and remove tuples from the pre-solution, to correctly simulate their effect the algorithm puts together different s-t tgds and uses negation to avoid the generation of unneeded tuples in the result. Other approaches and semantics for the rewriting of s-t tgds have also been recently introduced [1].

Another important aspect is the extension to nested relations and XML. The theoretical properties of data exchange in a general XML setting have been recently studied [3, 7], and, due to the generality, have been shown to exhibit several negative properties. However, important results were established for the fragment of XML data exchange in which the data model is restricted to correspond to nested relations. A very important result was reported in [7]: the authors show that the generation of universal solutions for a nested scenario can be reduced to the generation of solutions for a traditional, relational scenario, even in the presence of target constraints. The authors also provide an algorithm to perform the reduction.

We believe that these recent results, together with important theoretical studies, such as the ones on mapping invertion [11], open new possibilities for research on schema mappings. A notable example of a new generation of tool has been recently presented [15]. The ++Spicy tool can deal with different data management tasks, including data fusion, data cleaning and ETL scenarios, which, in our opinion, represent very promising areas of application of the latest schema-mappings and data-exchange techniques.

3 Conclusions

Schema mapping management is an important research area in data transformation, exchange and integration systems. From the early prototypes developed ten years ago, important results have been consolidated, but, despite the good results, the adoption of mapping systems in real-life integration applications has been slow. We have shown how emerging trends are overcoming the limits of the initial proposal and are going to encourage the developing of more systems based on schema mappings. On one side, novel theoretical results are paving the way to the creation of innovative applications for real world problems. On the other side, a new generation of tools for the creation and optimization of schema mappings are widening the opportunities offered by such technology.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

References

1. B. Alexe, M. A. Hernández, L. Popa, and W. C. Tan. Mapmerge: Correlating independent schema mappings. *PVLDB*, 3(1):81–92, 2010.
2. B. Alexe, W. Tan, and Y. Velegrakis. Comparing and Evaluating Mapping Systems with STBenchmark. *PVLDB*, 1(2):1468–1471, 2008.
3. M. Arenas and L. Libkin. XML Data Exchange: Consistency and Query Answering. *J. of the ACM*, 55(2):1–72, 2008.
4. C. Beeri and M. Vardi. A Proof Procedure for Data Dependencies. *J. of the ACM*, 31(4):718–741, 1984.
5. P. A. Bernstein and S. Melnik. Model Management 2.0: Manipulating Richer Mappings. In *SIGMOD*, pages 1–12, 2007.
6. A. Bonifati, G. Mecca, A. Pappalardo, S. Raunich, and G. Summa. Schema Mapping Verification: The Spicy Way. In *EDBT*, pages 85 – 96, 2008.
7. R. Chirkova, L. Libkin, and J. Reutter. Tractable XML Data Exchange via Relations. In *CIKM*, 2011.
8. R. Fagin, P. Kolaitis, R. Miller, and L. Popa. Data Exchange: Semantics and Query Answering. *TCS*, 336(1):89–124, 2005.
9. R. Fagin, P. Kolaitis, A. Nash, and L. Popa. Towards a Theory of Schema-Mapping Optimization. In *ACM PODS*, pages 33–42, 2008.
10. R. Fagin, P. Kolaitis, and L. Popa. Data Exchange: Getting to the Core. *ACM TODS*, 30(1):174–210, 2005.
11. R. Fagin, P. G. Kolaitis, L. Popa, and W. C. Tan. *Schema Matching and Mapping*, chapter Schema Mapping Evolution Through Composition and Inversion. 2011.
12. A. Fuxman, M. A. Hernández, C. T. Howard, R. J. Miller, P. Papotti, and L. Popa. Nested Mappings: Schema Mapping Reloaded. In *VLDB*, pages 67–78, 2006.
13. L. M. Haas, M. A. Hernández, H. Ho, L. Popa, and M. Roth. Clio Grows Up: from Research Prototype to Industrial Tool. In *SIGMOD*, pages 805–810, 2005.
14. B. Marnette, G. Mecca, and P. Papotti. Scalable data exchange with functional dependencies. *PVLDB*, 3(1):105–116, 2010.
15. B. Marnette, G. Mecca, P. Papotti, S. Raunich, and D. Santoro. ++Spicy: an opensource tool for second-generation schema mapping and data exchange. *PVLDB*, 4(11):1438–1441, 2011.
16. G. Mecca, P. Papotti, and S. Raunich. Core Schema Mappings. In *SIGMOD*, 2009.
17. R. J. Miller, L. M. Haas, and M. A. Hernandez. Schema Mapping as Query Discovery. In *VLDB*, pages 77–99, 2000.
18. R. Pichler and V. Savenkov. DEMo: Data Exchange Modeling Tool. *PVLDB*, 2(2):1606–1609, 2009.
19. L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernandez, and R. Fagin. Translating Web Data. In *VLDB*, pages 598–609, 2002.
20. L. Seligman, P. Mork, A. Halevy, K. Smith, M. J. Carey, K. Chen, C. Wolf, J. Madhavan, A. Kannan, and D. Burdick. OpenII: an Open Source Information Integration Toolkit. In *SIGMOD*, pages 1057–1060, 2010.
21. B. ten Cate, L. Chiticariu, P. Kolaitis, and W. C. Tan. Laconic Schema Mappings: Computing Core Universal Solutions by Means of SQL Queries. *PVLDB*, 2(1):1006–1017, 2009.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A Linear Algebra Approach for Supply Chain Management*

Roberto De Virgilio and Franco Milicchio

Dipartimento di Informatica e Automazione
Università Roma Tre, Rome, Italy
{dvr,milicchio}@dia.uniroma3.it

Abstract. In current trends of consumer products market, the amount of RFID data in supply chain management is vast, posing significant challenges for attaining acceptable performance on their analysis. Current approaches provide hard-coded solutions, with high consumption of resources; moreover, these exhibit very limited flexibility dealing with multidimensional queries, at various levels of granularity and complexity. In this paper we propose a general model for supply chain management based on the first principles of linear algebra, in particular on *tensorial calculus*. Leveraging our abstract algebraic framework, our technique allows both quick decentralized on-line processing, and centralized off-line massive business logic analysis, according to needs and requirements of supply chain actors. Experimental results show that our approach, utilizing recent linear algebra techniques can process analysis efficiently.

1 Introduction

In the management of a supply chain, main retailers are investing in new technologies in order to boost the information exchange. To this aim, RFID, the Radio-Frequency Identification, is a recent potential wireless technology. An RFID application usually generates a stream of tuples, usually called *raw data*, of the form of a triple (e, l, t), where e is an EPC, l represents the location where an RFID reader has scanned the e object, and t is the time when the reading took place. Other properties can be retrieved, *e.g.*, temperature, pressure, and humidity. A single tag may have multiple readings at the same location, thus potentially generating an immense amount of raw data. Therefore, a simple data cleaning technique consists in converting raw data in *stay records* of the form (e, l, t_i, t_o), where t_i and t_o are the time when an object enters or leaves a location l, respectively. In this manuscript, we address the challenging problem of efficiently managing the tera-scale amount of data per day, generated by RFID applications (cf. [1], and [2,6,7]), focusing on stay records as the basic block to store RFID data.

* This is the extended abstract of a paper that appears in the Proceedings of CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Contribution. Leveraging such background, this paper proposes a general model of supply chains, mirrored with a formal *tensor* representation (*i.e.* a generalization of linear forms, usually represented by matrices) and endowed with specific operators, allowing both quick decentralized on-line processing, and centralized off-line massive business logic analysis, according to needs and requirements of supply chain actors.

Outline. Our manuscript is organized as follows. In Section 2 we will briefly recall the available literature. The general supply chain model, accompanied by a formal tensorial representation is supplied in Section 3, subsequently put into practice in Section 4, where we provide the reader a method of analyzing RFID data within our framework. We benchmark our approach with several test beds, in Section 5, while Section 6 sketches conclusion and future work.

2 Related Work

In a real scenario, great lapse and huge amounts of data are generated. To this aim, knowledge representation techniques focus on operating deep analysis in such systems. An alternative approach [6] consists in warehousing RFID data and performing multidimensional analyses on the warehouse. The focus here is on data compression techniques and on storage models with the goal of achieving a more expressive and effective representation of RFID data. A straightforward method is to provide a support to path queries (*e.g.*, find the average time for products to go from factories to stores in Seattle) by collecting RFID tag movements along the supply chain. Usually, the tag identifier, the location and the time of each RFID reading is gathered and stored in a huge relational table. Lee et al. have proposed an effective path encoding approach to represent the data flow representing the movements of products [7]. In a path, a prime number is assigned to each node and a path is encoded as the product of the number associated with nodes. Mathematical properties of prime numbers guarantee the efficient access to paths. A major limitation of the majority of these approaches have to fix the dimensions of analysis in advance to exploit *ad-hoc* data structures to be maintained. It follows that, in many application scenario the compression loses its effectiveness and the size of tables does not be reduced significantly. Therefore we have a limited flexibility when multidimensional queries, at varying levels of granularity and complexity, need to be performed.

3 RFID data Modeling

This section is devoted to the definition of a general model capable of representing all aspects of a given supply chain. Our overall objective is to give a rigorous definition of a supply chain, along with few significant properties, and show how such representation is mapped within a standard tensorial framework.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

3.1 A General Model

Let us define the set $\mathcal{E}$ as the set of all EPCs, with $\mathcal{E}$ being finite. A *property* of an EPC is defined as an application $\pi : \mathcal{E} \to \Pi$, where Π represents a suitable property codomain. Therefore, we define the application of a property $\pi(e) := \langle \pi, e \rangle$, *i.e.*, a property related to an EPC $e \in \mathcal{E}$ is defined by means of the pairing EPC-property; a property is a surjective mapping between an EPC and its corresponding property value. A *supply chain* is defined as the product set of all EPCs, and all the associated properties. Formally, let us introduce the family of properties π_i, $i = 1, \ldots, k + d < \infty$, and their corresponding sets Π_i; we may therefore model a supply chain as the product set

$$\mathcal{S} = \mathcal{E} \times \Pi_1 \times \ldots \times \Pi_{k-1} \times \Pi_k \times \ldots \times \Pi_{k+d} \,. \tag{1}$$

3.2 Properties

In the following we will focus on some of the properties related to EPCs, *i.e.*, we will model some codomains Π and their associated features.

Location. Let us briefly model the *location* associated to an EPC. It is common to employ a GPS system in order to track the position on earth, however, any fine-grained space is sufficient to our purposes. In particular, being the earth homeomorphic to a 3-sphere, any ordered triple of real numbers suffices, leading us to the following definition:

Definition 1 (Location) *Let $\mathcal{L}$ be the set of ordered tuples $\ell := (\ell_1, \ell_2, \ell_3)$, with $\ell_1, \ell_2, \ell_3 \in \mathbb{R}$. We name $\mathcal{L}$ as* location set*, ℓ_1, ℓ_2 and ℓ_3 as* location coordinates*.*

Time. In order to model a temporal interval relative to a product (EPC), we resort to an ordered couple of elements from the ring of real numbers. This suffices to specify, *e.g.*, the *entry* and *exit* time of a product from a given location. With this point of view, we model time as follows:

Definition 2 (Time) *Let $\mathcal{T}$ be the set of ordered couples $\tau := (t_i, t_o)$, with $t_i, t_o \in \mathbb{R}$. We name $\mathcal{T}$ as* time set*, t_i and t_o as* incoming *and* outcoming timestamps*, respectively.*

Definition 3 (Inner sum) *Let us define the* inner sum *of two time elements $\tau_1 = (t_i^1, t_o^1)$, $\tau_2 = (t_i^2, t_o^2)$ as the operator $\oplus : \mathcal{T} \times \mathcal{T} \to \mathcal{T}$:*

$$\tau_1 \oplus \tau_2 := \left(\min(t_i^1,\, t_i^2), \max(t_o^1,\, t_o^2) \right) \,.$$

With the above operator, we have that $(\mathcal{T}, \oplus)$ assumes the algebraic structure of an abelian group. Such operation allows us to rigorously model the "addition of products": the overall timeframe of two products is, in fact,

Definition 4 (Lifetime) *We define as* lifetime *the linear form $\lambda : \mathcal{T} \to \mathbb{R}$ defined as follows: $\langle \lambda,\, \tau \rangle := t_o - t_i \,, \quad \tau = (t_i, t_o) \in \mathcal{T}$.*

Due to the linearity, we are allowed to construct equivalence classes in $\mathcal{T}$ as

$$[\widetilde{\tau}] := \{ \tau \in \mathcal{T} : \langle \lambda,\, \tau \rangle = \langle \lambda,\, \widetilde{\tau} \rangle \} \,. \tag{2}$$

The *canonical representative elements* of the above equivalence classes are defined as $(0, t_o)$, with $t_o \in \mathbb{R}$.

Definition 5 (Admissible time) *Given a time element $\tau \in \mathcal{T}$, we say that $\tau = (t_i, t_o)$ is* admissible *iff $\langle \lambda,\, \tau \rangle > 0$, with $t_i, t_o > 0$.*

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

3.3 Tensorial Representation

Let us now introduce a formal *tensorial framework* capable of grasping all properties related to a supply chain, as proposed in Section 3.1. We divide properties into two categories, *countable* and *uncountable* spaces. This separation allows us to represent countable spaces with natural numbers, therefore mapping their product space to $\mathbb{N}^k$, while leaving the product space of all uncountable properties into a collective space $\mathbb{U}$:

$$\mathcal{S} = \underbrace{\mathcal{E} \times \Pi_1 \times \ldots \times \Pi_{k-1}}_{\mathbb{N}^k} \times \underbrace{\Pi_k \times \ldots \times \Pi_{k+d}}_{\mathbb{U}} . \tag{3}$$

Such mapping will therefore introduce a family of injective functions called *indexes*, defined as: $\mathrm{idx}_i : \Pi_i \longrightarrow \mathbb{N}$ with $i = 1, \ldots, k-1$. When considering the set $\mathcal{E}$, we additionally define a supplemental index, the *EPC index function* $\mathrm{idx}_0 : \mathcal{E} \to \mathbb{N}$, consequently completing the map of all countable sets of a supply chain $\mathcal{S}$ to natural numbers.

Definition 6 (Tensorial Representation) *The* tensorial representation *of a supply chain $\mathcal{S}$, as introduced in equation* (1)*, with countability mapping as in* (3) *is a multilinear form $\Sigma : \mathbb{N}^k \longrightarrow \mathbb{U}$.*

A supply chain can be therefore rigorously denoted as a rank-k tensor with values in $\mathbb{U}$, mapping countable to uncountable product space.

3.4 Implementation

Preliminary to describing an implementation of our supply chain model, based on tensorial algebra, we pose our attention on the practical nature of a supply chain. Our treatment is general, representing a supply chain with a tensor, *i.e.*, with a multidimensional matrix. Supply chain rarely exhibit completion: practical evidence [5] suggests that products, identified by their EPC, for example, seldom present themselves in every location. The same consideration applies also to other properties, in particular, to countable properties. Hence, our matrix effectively requires to store only the information regarding connected nodes in the graph: as a consequence, we are considering sparse matrices [3], *i.e.*, matrices storing only non-zero elements.

Example 1. Let us consider the supply chain pictured in Fig. 1, described tensorially by $\Sigma : \mathbb{N}^2 \to \mathcal{T}$, whose representative matrix is as follows:

$$\begin{pmatrix}
(0,2) & \cdot & \cdot & (5,7) & \cdot & (8,10) & \cdot & \cdot \\
\cdot & \cdot & (0,3) & \cdot & \cdot & \cdot & \cdot & (4,9) \\
(0,1) & \cdot & \cdot & (4,5) & \cdot & \cdot & (6,9) & \cdot \\
\cdot & (0,4) & \cdot & \cdot & (5,6) & \cdot & \cdot & (8,9) \\
\cdot & \cdot & (0,5) & \cdot & (6,7) & \cdot & \cdot & (9,11)
\end{pmatrix}$$

where, for typographical simplicity, we omitted $0_{\mathcal{T}} = (0,0)$, denoted with a dot. For clarity's sake, we outline the fact that Σ is a rank-2 tensor with dimensions 5 (i.e. the rows), 8 (i.e. the columns). In fact, we have

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

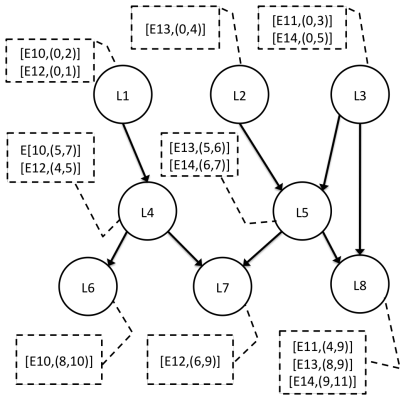


Fig. 1. An example supply chain represented by a directed graph. Callouts represent lists of tuple constituted by an EPC with the associated time (t_i, t_o).

$$\mathcal{E} = \{E10, E11, E12, E13, E14\} \ ,$$
$$\mathit{\Pi}_1 = \{L1, L2, L3, L4, L5, L6, L7, L8\} \ ,$$

where $\mathrm{idx}_0(E10) = 1$, $\mathrm{idx}_0(E11) = 2, \ldots, \mathrm{idx}_0(E14) = 5$, and similarly $\mathrm{idx}_1(L1) = 1, \ldots, \mathrm{idx}_1(L8) = 8$. In the sparse representation we have $\{\{1,1\} \to (0,2), \{1,4\} \to (5,7), \ldots, \{5,8\} \to (9,11)\}$.

4 RFID data Analysis

Based on our conceptual framework, in this section we will provide a method to analyze RFID data represented by a tensor.

Tracking Query. A *tracking query* finds the movement history for a given tag identifier $e \in \mathcal{E}$. We can comfortably perform the query efficiently, using the model described in Section 3, by applying the tensor application. Therefore, given $i = \mathrm{idx}(e)$, we build a Kroneker vector as a vector δ_i, with $|\delta_i| = |\mathcal{E}|$, and finally apply of the rank-2 tensor represented by M_{ij} to δ_i: $r = M_{ij}\delta_i$. For instance, referring to the example pictured in Fig. 1, let us consider the tag $E13$, we have $i = \mathrm{idx}(E13) = 4$, and therefore our vector will be $\delta_4 = \{\{4\} \to 1\}$. Consequently, the resulting vector will be $r = M_{ij}\delta_4 = \{\{2\} \to (0,4),\ \{5\} \to (5,6),\ \{8\} \to (8,9)\}$ or in another notation, $L2 \to L5 \to L8$.

Path Oriented Query. A *path oriented query* returns the set of tag identifiers that satisfy different conditions. Following the query templates given in [7], we subdivide path oriented queries into two main categories: *path oriented retrieval* and *path oriented aggregate* queries. The former returns all tags covering a path satisfying given conditions, while the latter computes an aggregate value. It is possible to formulate a grammar for these queries, similar to XPath expressions; in particular, path oriented retrieval and path oriented aggregate queries are respectively indicated as follows: $L_1[\mathrm{cond}_1]//L_2[\mathrm{cond}_2]//\ldots//L_n[\mathrm{cond}_n]$ or

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

$L_1[\text{cond}_1]/L_2[\text{cond}_2]/\ldots/L_n[\text{cond}_n]$, where $L_1, \ldots L_n$ is the *path condition*, *i.e.*, the sequence of locations covered by the tag, and cond_i is the *info condition*; for more information about such expressions, we refer the reader to [7]. A path condition expresses *parenthood* between locations, indicated as L_i/L_j, or *ancestry* with $L_i//L_j$; an info condition, on the other hand, indicates conditions on tag properties, *e.g.*, $StartTime$ and $EndTime$.

In our framework, a *path oriented retrieval query* is easily performed by exploiting the tensor application coupled with the Hadamard product. Given the location set $\mathcal{L}$, for each $z = \text{idx}(L_k)$, with $k = 1, \ldots, n$, we create a Kroneker vector δ_z and subsequently apply the rank-2 tensor, resulting in a set of vectors $r_z = M_{ij}\delta_j$, where for typographical reasons, we dropped the subscript intending $\delta \equiv \delta_z$. Finally, we apply the condition function to each r_z employing the map operator. This yields a set of vectors

$$\widetilde{r_z} = \text{map}(\text{cond}_z, r_z)\,, \qquad \text{cond}_z : \mathbb{N}^k \times \mathbb{U} \longrightarrow \mathbb{F}\,,$$

whose Hadamard multiplication generates the final result: $\bar{r} = \widetilde{r_1} \circ \ldots \circ \widetilde{r_n}$, reminding the reader that only non-zero values are stored, and therefore given as a result of a computation. The cond functions, as indicated above, are maps between properties of supply chains and a suitable space $\mathbb{F}$, *e.g.*, natural numbers for a boolean result. For an example, referring to Fig. 1, let us consider the query

$$L3[StartTime > 0]//L8[EndTime - StartTime < 4]\,.$$

In this case, given $\text{idx}(L3) = 3$ and $\text{idx}(L8) = 8$, we build δ_3 and δ_8, and generate the partial results : $r_3 = M_{ij}\delta_3 = \{\{2\} \to (0, 3), \{5\} \to (0, 5)\}$ and $r_8 = M_{ij}\delta_8 = \{\{2\} \to (4, 9), \{4\} \to (8, 9), \{5\} \to (9, 11)\}$, where evidently the application was performed along the second dimension, *i.e.*, for each $\delta \in \{\delta_3, \delta_8\}$, we compute $M_{ij}\delta_j$. Finally, subsequent to mapping conditions on the results, in this case $\text{cond}_3 = t_i(\cdot) > 0$ and $\text{cond}_8 = \langle \lambda, \cdot \rangle < 4$, we obtain the correct outcome $\bar{r} = \widetilde{r_3} \circ \widetilde{r_8} = \{\{5\} \to (9, 11)\}$, *i.e.*, $E14$.

Considering parenthood instead of ancestry, *i.e.*, L_i/L_j, we briefly sketch the fact that such query does not, in fact, differ from the above, except for one particular: each resulting EPC, when subject to a tracking query, produces a sparse vector whose *length*, *i.e.*, the number of non-zero stored elements, is exactly equal to the number of locations under analysis.

Path oriented aggregate queries may be represented as $\langle f, Q \rangle =: f(Q)$ where f is an *aggregate function*, *e.g.*, average or minimum, and Q is the result of a path oriented retrieval query. Therefore, let $r_{\bar{Q}}$ be the result of Q, and let f be a function defined on vectors of supply chain elements, we simply have that a path aggregate may be expressed as $f(r_{\bar{Q}})$. Referring to Fig. 1, let us consider the query expressed in the grammar of [7]: $\langle AVG[L8.StartTime], //L8 \rangle$. It is easily performed by applying the function $f := \text{average}(t_i)$ to the outcomes of the path oriented retrieval query on $L8$, resulting in $\bar{r} = \text{average}(\{4, 8, 9\}) = 7$.

5 Experiments

We performed a series of experiments aimed at evaluating the performance of our approach, reporting the main results in the present section.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Environment. Our benchmarking system is a dual core 2.66 GHz Intel with 2 GB of main memory running on Linux, where we implemented our framework[1] in C++ within the Mathematica 8.0 computational environment. Our results have been compared to the ones from the approach in [7], tested against a generated synthetic RFID data in terms of stay records, and considering products moving together in small groups or individually, a behavior called *IData* in [7]: data production followed the same guidelines on a supply chain of 100 locations. Finally the complete data set comprises 10^5, $5 \cdot 10^5$, 10^6, $5 \cdot 10^6$, and 10^7 stay records. In the following, we will denote our tensorial approach T, while the proposed one in [7] with P.

Results. Performances have been measured with respect to data loading and querying. Referring to the former, the main advantage of our approach is that we are able to perform loading without any particular relational schema, when compared to P, where a schema coupled with appropriate indexes have to be maintained by the system. In this case, loading execution times are 0.9, 11, and 113 seconds, for sets of 10^5, 10^6, and 10^7 stay records, respectively; on the contrary, P timings were in order of minutes and hours. Another significant advantage of T relies in memory consumption: we need 13, 184, and 1450 MB to import the above mentioned sets; as a side-note, we highlight the fact that the 10^7 set required a division in smaller blocks, *e.g.*, 10^6, due to the limited memory at disposal. With respect to query execution, T presents a similar behavior and advantages with respect to P, for both time and memory consumption. We performed *cold-cache* experiments, *i.e.*, dropping all file-system caches before restarting the systems and running the queries, and repeated all the tests three times, reporting the average execution time. As in [7], we formulated 12 queries to test the two systems as reported in [4]. In brief, Q1 is a tracking query, Q2 to Q5 are path oriented retrieval queries, while Q6 to Q12 are path oriented aggregate queries. Due to the nature of P, we were able to perform a comparison only on centralized off-line massive analysis. A significant result is the *speed-up* between the two approaches, as shown in Fig. 2.(a); again, Q1 label is dropped due to typographical causes. We computed the speed-up for all data sets as the ratio between the execution time of P, and that of our approach T, or briefly $S = {}^{t_P}/_{t_T}$. In general, T performs very well with respect to P in any dataset, particularly for queries related to the object transition, *e.g.*, Q2, Q4, Q5, Q10. The query performance of T is on the average 19 times better than that of P, 150 times on the maximum, *i.e.*, Q10. Another strong point of T is a very low consumption of memory, due to the *sparse matrix* representation of tensors and vectors. Fig. 2.(b) illustrates the main memory consumption of each query with respect to 10^5, 10^6, and 10^7 stay records. On the average, tracking queries require very few bytes of memory for any dataset, path oriented queries few KBytes, topping 1 MB for 10^7. Results demonstrate how our approach can be used in a wide range of applications, where devices with limited calculus resources may process large amount of data in an efficient and effective way.

[1] A prototype implementation is available at http://pamir.dia.uniroma3.it:8080/SimpleWebMathematica

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

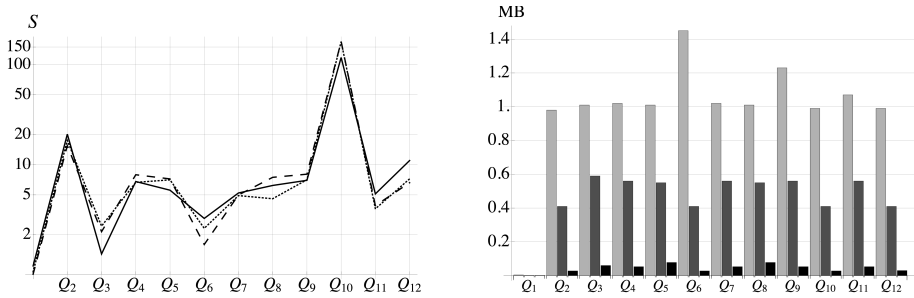


Fig. 2. (a) Speed-up logarithmic graph for all queries with 10^5 (solid), 10^6 (dashed), and 10^7 (dotted). (b) Main memory consumption for each query: black bars refer to 10^5, dark gray to 10^6, and light gray to 10^7 data size.

6 Conclusion and Future Work

We have presented an abstract algebraic framework for the efficient and effective analysis of RFID data in supply chain management. Our approach leverages tensorial calculus, proposing a general model that exhibits a great flexibility with multidimensional queries, at diverse granularity and complexity levels. Experimental results proved our method efficient when compared to recent approaches, yielding the requested outcomes in memory constrained architectures. For future developments we are investigating the introduction of reasoning capabilities, along with a thorough deployment in highly distributed Grid environments. In addition, we are about to test our model on mobile devices, comprising more complex properties and queries.

References

1. Angeles, R.: Rfid technologies: supply-chain applications and implementation issues. Information Systems Management 22(1), 51–65 (2005)
2. Bai, Y., Wang, F., Liu, P., Zaniolo, C., Liu, S.: Rfid data processing with a data stream query language. In: ICDE (2007)
3. Davis, T.A.: Direct Methods for Sparse Linear Systems. SIAM (2006)
4. De Virgilio, R., Milicchio, F.: Tensor calculus for rfid data management. Tech. Rep. RT-DIA-182, University Roma Tre (2011)
5. Derakhshan, R., Orlowska, M.E., Li, X.: Rfid data management: Challenges and opportunities. In: RFID. pp. 175–182 (2007)
6. Gonzalez, H., Han, J., Li, X., Klabjan, D.: Warehousing and analyzing massive rfid data sets. In: ICDE (2006)
7. Lee, C.H., Chung, C.W.: Efficient storage scheme and query processing for supply chain management using rfid. In: SIGMOD. pp. 291–302 (2008)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A Semantic Method for Searching Knowledge in a Software Development Context*

Sonia Bergamaschi, Riccardo Martoglia, and Serena Sorrentino

DII, University of Modena and Reggio Emilia
via Vignolese 905, 41125 Modena, Italy
firstname.lastname@unimore.it

Abstract. The FACIT-SME European FP-7 project targets to facilitate the use and sharing of Software Engineering (SE) methods and best practices among software developing SMEs. In this context, we present an automatic semantic document searching method based on Word Sense Disambiguation which exploits both syntactic and semantic information provided by external dictionaries and is easily applicable for any SME.

1 Introduction and Motivation

Over the last years, in Europe, software development is becoming a bottleneck in the development of the Information Society, especially for SMEs (Small and Medium Enterprises) which need to allocate mostly all of their available resources on its production rather than on new technology training. The main goal of the European FP7 3 years project "Facilitate IT-providing SMEs by Operation-related Models and Methods (FACIT-SME)" is to facilitate IT SMEs in sharing and (re)using SE methods, tools, and experiences for systematically designing and developing their applications integrated with the business processes. In order to achieve this goal, the project proposes a novel Open Reference Model (ORM) [4] serving as an underlying knowledge backbone which stores existing reference knowledge for software-developing SMEs, including different engineering methods, tools, quality model requirements, and enterprise model fragments of IT SMEs in a computer-processable form. On top of the ORM repository, a customizable Open Source Enactment System (OSES) [3] provides IT support for the project-specific application of the ORM. As key part of the OSES, specific query-based search methods support the organizations in: finding a new methodology, by selecting ORM elements that best match given specific enterprise objectives (i.e., "From Scratch" scenario); improving a given existing methodology, suggesting the ORM information most relevant to it (i.e., "From methodology" scenario).

* This extended abstract summarizes the research work we performed in the first two years of the FACIT-SME project, including a summarization of the preliminary results described in [12] (SEKE 2011). It was partially supported by the European Community's Seventh Framework Programme managed by REA Research Executive Agency (http://ec.europa.eu/research/rea)([FP7/2007-2013][FP7/2007 - 2011])

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

In our research work, we focus on search/filtering methods by taking advantage of the textual information (which we will refer to as *documents*) stored in the ORM and/or already available in each enterprise. In this context, queries are provided in textual form, e.g. keywords/sentences about the company background and project requirements, or even existing methodology descriptions (for the second scenario). Standard search methods based on syntactic techniques [5] are often inadequate to capture the similarity between documents, as they do not consider the semantics associated with the terms composing documents. For instance, without exploiting semantics, i.e., synonyms and related terms, the piece of document *D1* "*...**clients** for your small business enterprise...*" would wrongly be deemed as irrelevant to the query fragment *Q1 "....product requirements specified by the **customer**..."*. Moreover, terms might be *ambiguous*, i.e., they may have more than one possible meaning. For instance, even if the piece of document *D2* "*Distributed applications partition workloads between servers and **clients**...*" contains "client", the term is used in a completely different context, thus it should not be presented among the results.

In this paper, we propose a semantic method, implemented in the **Semantic Helper** component of the FACIT-SME solution, for searching ORM documents. It exploits a standard information retrieval weighting/ranking scheme extended to take into account synonyms and related terms information, together with Word Sense Disambiguation (WSD) techniques, and leads to the following achievements: (1) it is a fully automatic and semantic method that overcomes the standard syntactic technique limitations; (2) it is devised for IT SMEs, providing them with a flexible and easy-to-apply method that does not require big investments or knowledge prerequisites.

The rest of the paper is organized as follows: in Sections 2, we describe the Semantic Helper and the phases of the processes it supports; in Section 3, we describe the experimental evaluation of our method, while Section 4 concludes the work and briefly analyzes related works.

2 The Semantic Helper

The Semantic Helper supports the FACIT-SME solution by performing two main processes (see Figure 1):

1. **Semantic Glossary Population:** during this off-line process, statistical and semantic information are automatically extracted from the ORM documents and stored in a repository called Semantic Glossary;
2. **Relevant Document Ranking/Retrieval:** in this online process, user queries are processed and relevant documents are identified by exploiting the information provided by the Semantic Glossary.

In these two processes, we can identify three main phases: (a) keyword extraction and enrichment; (b) Semantic Glossary generation; (c) semantic similarity computation.

Keyword Extraction and Enrichment. The goal of this phase (involved in both processes) is to automatically extract, normalize and disambiguate terms

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

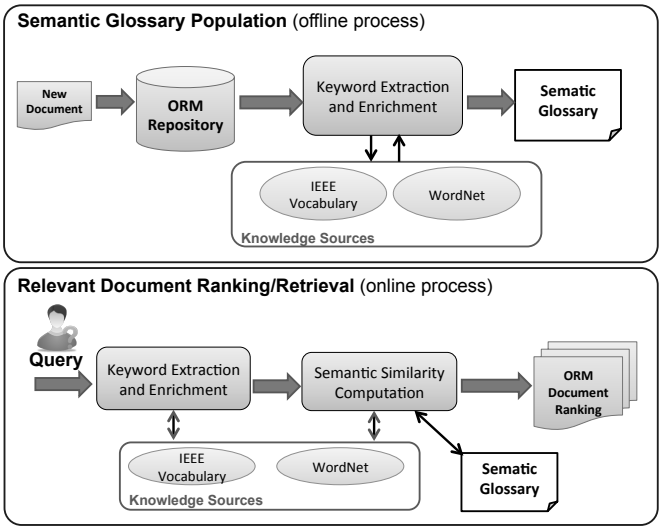


Fig. 1. Semantic Helper offline (top part) and online (bottom part) processes.

from the ORM documents collection/user queries. First of all, terms are extracted and normalized by means of *tokenization*, *stemming*, and *Part of Speech (POS) tagging*; moreover, possible *composite terms* (such as "product action plan" or "product requirements") are identified by means of a simple state machine and of POS tags information. Then, by exploiting external knowledge sources, the most relevant terms are selected. Finally, the selected terms[1] are annotated through a WSD algorithm, and associated with additional information such as definitions and synonyms.

The Semantic Helper makes use of two knowledge sources: the IEEE Software and Systems Engineering Vocabulary[2], and the WordNet[3] English thesaurus. These two sources differ in *Coverage*: the IEEE vocabulary includes only specialist terms in the project and computer science area, while WordNet complements it with general knowledge about English concepts; and *Granularity*: the IEEE vocabulary makes a very fine-grained sense distinction while WordNet makes coarser-grained sense distinctions. As in the FACIT-SME project we do not need a fine sense distinction (as errors may typically come out when terms have orthogonal senses), we employ WSD only for terms existing in WordNet (terms not existing in WordNet but in the IEEE vocabulary are associated with the first sense proposed).

To perform WSD, we use the STRIDER WSD algorithm described in [11]; however, other WSD algorithms might be employed as the ones described in [13, 14]. Given a document D containing a set of terms $\{t_1, t_2, ..., t_n\}$, for each term

[1] We limit the selection to nouns, as most of the semantics of a sentence is usually carried by noun terms [13].

[2] `http://www.computer.org/sevocab`

[3] `http://wordnet.princeton.edu/`

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

ANNOTATION	WN	IEEE	SYNS	DEFS	IDF	DOC_LIST
business enterprise#1	Y	-	commercial_enterprise#2, business#2	the activity of providing goods and services involving financial and commercial and industrial aspects.	0,6931	['D1']
client#3	Y	-	guest#4, node#7	any computer that is hooked up to a computer network; etc.	0,9315	['D2']
customer#1	Y	-	client#2	someone who pays for goods or services	0,8324	['D1']
…	…	…	…	…	…	…

Fig. 2. An excerpt of the Sense-Aware Semantic Glossary extracted from *D1* and *D2* (global view).

t_i, where $t_i \in D$, STRIDER returns an annotation, $A(t_i) = s_j$, where s_j is the top sense of a ranking $\{s_j, ..., s_k\}$ of plausible senses for t_i.

Semantic Glossary Generation. The Semantic Glossary is created the first time in the "offline" process, with all the documents available in the ORM; then it is automatically updated/enriched in case of new content is added to the ORM. It consists of a **global view** (all annotated terms in all documents, together with their statistics, see Figure 2), where each entry includes:

ANNOTATION: the annotation information in the form *"*$term_{\#senseIndex}$*"* (e.g., "$client_{\#3}$", where "client" has been annotated with the third sense proposed by the knowledge source);
WN and IEEE: if the annotation is w.r.t. WN or the IEEE vocabulary;
SYNS: possible synonym annotations;
DEFS: the definition corresponding to the annotation;
IDF: the annotation inverse document frequency [15] in the collection;
DOC_LIST: documents in which the annotation occurs.

and a **per-document view** (the annotated terms occurrences in each document with their statistics), containing:

DOC: the document ID in which the annotation occurs;
ANNOTATION: the annotation in the form *"*$term_{\#senseIndex}$*"*;
AF: the frequency of this annotation in the document, normalized by the total number of annotations in the document;
WEIGHT: the AF*IDF weight of the annotation.

Semantic Similarity Computation. The need of effectively and efficiently computing similarities between ORM/query documents is crucial to the project. To this end, we defined a *document similarity formula* $DSim(D^x, D^y)$ which, given a source document $D^x = \{t_1^x, ..., t_n^x\}$ (e.g., a given quality requirement description) and a target document $D^y = \{t_1^y, ..., t_n^y\}$ (e.g., each software methodology description available in the ORM), quantifies the similarity of the source document w.r.t. the target document. The computation of $DSim$ involves considering each annotation in D^x and finding the most similar annotation available in D^y by exploiting a *sense similarity formula* $SSim$. Thus, a **ranking** of the available documents is induced, predicting which documents are relevant and which are not w.r.t. D^x (i.e., the query document).

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Equation (1) shows the document similarity formula $DSim(D^x, D^y)$ which is given by the sum of all sense similarities $SSim$ between each annotation in D^x and the annotation (defined in (2)) in D^y maximizing the sense similarity with the annotated term in D^x:

$$DSim(D^x, D^y) = \sum_{t_i^x \in D^x} SSim(A(t_i^x), A(t^y_{\overrightarrow{j}(i)})) \cdot w_i^x \cdot w^y_{\overrightarrow{j}(i)} \quad (1)$$

$$A(t^y_{\overrightarrow{j}(i)}) = argmax_{t_j^y \in D^y}(SSim(A(t_i^x), A(t_j^y))) \quad (2)$$

where $A(t_i^x)$, $A(t_j^y)$ are the annotations of the i-th (j-th) term in the document x (y), respectively, and w_i^x and $w^y_{\overrightarrow{j}(i)}$ are the weights associated with the annotations and computed in the pre-document view of the Semantic Glossary. By exploiting weights common annotations, which are probably less meaningful, will give a lower contribute to the final similarity. $SSim$ is a sense similarity function which computes the similarity between two annotations:

$$SSim(A(t_i), A(t_j)) = \begin{cases} 1, & \text{if } A(t_i)\ SYN\ A(t_j) \\ r, & \text{if } A(t_i)\ REL\ A(t_j) \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where the case of maximum similarity (value 1) corresponds to the case where the two annotations are synonyms (SYN relation). Moreover, the formula provides a further case where the two annotations are in some way related (REL relation) from a semantic point of view (i.e., broader/narrower sense etc.): such annotations will contribute with a similarity of r, where $0 < r < 1$. While the synonym information straightly comes from the Semantic Glossary, equation (4) shows a possible way of computing the similarity by exploiting the *glosses* (definitions) of the senses:

$$GSim(gl(A(t_i)), gl(A(t_j))) = \sum |ovl(gl(A(t_i)), gl(A(t_j)))|^2 \quad (4)$$

where $gl(A(t))$ is the gloss associated with the annotation of t. In this case, two annotations are semantically related if their *gloss similarity* $GSim$ (which quantifies the similarity by finding overlaps ovl between the two glosses [6]) exceeds a given threshold Th.

We also provide a further way of computing semantic relatedness by exploiting the relations between terms coming from the WordNet thesaurus [12]. Indeed, in WordNet senses (synsets) are connected to other synsets by hypernym (i.e., "is-a") relations. Two annotations (i.e., senses) are semantically related if their *hypernym similarity* $HSim$, which is defined as inversely proportional to the length of the shortest path connecting them (as in many knowledge management works [9, 11]), exceeds a given threshold Th.

Example. As simple summarizing example, let us consider again the pieces of documents $D1$ and $D2$ and the query fragment shown in Section 1. After the application of the keyword extraction and enrichment phase to $D1$ and $D2$, we obtain the Semantic Glossary (partially) shown in Figure 2. Thus, at query time, we need to apply WSD only to the terms extracted from the user query $Q1$. By computing the document similarity between $Q1$, $D1$ and $Q1$, $D2$, we obtain:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Query	Semantic			No WSD			Syntactic (no syn/rel) (baselines)		
	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F
Q1	0,968	1,000	0,984	0,947	1,000	0,973	1,000	0,079	0,146
Q2	0,901	1,000	0,948	0,878	1,000	0,935	1,000	0,077	0,143
Q3	0,937	0,946	0,941	0,923	0,949	0,936	1,000	0,333	0,500
Q4	1	0,828	0,906	0,839	0,837	0,838	1,000	0,642	0,782
Q5	0,982	0,754	0,853	0,313	0,781	0,447	0,712	0,303	0,425
Q6	0,961	0,634	0,764	0,203	0,688	0,313	0,652	0,043	0,081

Fig. 3. Effectiveness analysis: precision, recall and F-measure (standard results for all-senses techniques on the left, two baselines on the right).

$$DSim(Q1, D1) = \sum_{t_i^{Q1} \in Q1} SSim(A(t_i^{Q1}), A(t_{\bar{j}(i)}^{D1})) \cdot w_i^{Q1} \cdot w_{\bar{j}(i)}^{D1} = 0.8$$

$$DSim(Q1, D2) = \sum_{t_i^{Q1} \in Q1} SSim(A(t_i^{Q1}), A(t_{\bar{j}(i)}^{D2})) \cdot w_i^{Q1} \cdot w_{\bar{j}(i)}^{D2} = 0$$

i.e., the document similarity between $Q1$ and $D2$ is equal to 0, and only the document $D1$ is returned as relevant for the query.

3 Experimental Evaluation

In this section, we present the results of the effectiveness evaluation we performed, on the proposed method, in the context of FACIT-SME. We formed a collection of 1500 documents (i.e., textual descriptions) about quality requirements taken from the ORM and derived from existing quality models, such as CMMI [2] and ISO 9000 [1]. Starting from this collection, we automatically generated the Semantic Glossary and we considered different queries simulating possible "From Scratch" scenario requests. Among them, we selected a set of 6 queries ($Q1 - Q6$) as the most representative ones: while queries $Q1 - Q3$ are mostly constituted by specific domain terms, queries $Q4 - Q6$ are representative of less common but possibly more complex to be handled requests, since they also contain several common and potentially more ambiguous terms.

For each query, we compared the output of the Semantic Helper with a "gold standard", i.e. relevant answers manually selected from the collection by experts in the field, and assessed precision, recall, and F-measure (Figure 3, left part). In the right part of the figure, we also present two baselines, i.e., the proposed semantic method working at term-level rather than at sense-level (no WSD applied, as in [12]), and a syntactic retrieval method ignoring synonyms and related terms, representative of document retrieval techniques commonly exploited by commercial systems. As we can see, the precision and recall levels achieved by the semantic method are generally very satisfying: all queries greatly benefit from semantic features such as synonyms and related terms management.We also notice that, due to the very specific and technical nature of the terms in $Q1 - Q3$, such queries are generally not largely affected by disambiguation issues.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

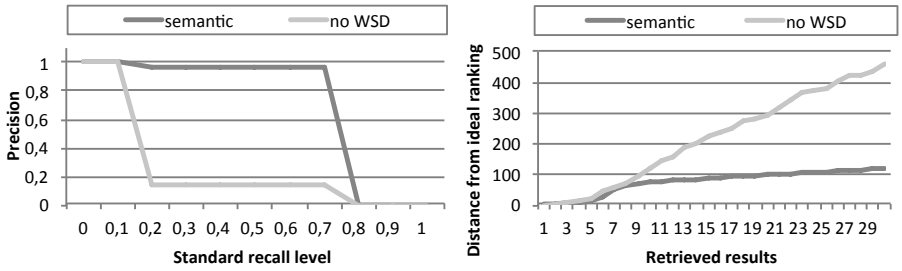


Fig. 4. In-depth effectiveness analysis for QT1: precision at standard recall levels (left) and distance from optimal ranking (right)

On the other hand, queries $Q4 - Q6$ contain several common and potentially ambiguous terms that are not present in the specialized IEEE vocabulary and for which only the use of WordNet similarity is allowed. This leads non-WSD techniques to several false-positives, especially in Q5 and Q6, while only the complete semantic method achieves satisfying precision/recall levels (at least more than double F measures in Figure 3). For instance, query $Q6$ contains, among others, the term "area": ORM contains few documents about "area" as "geographical region", while it contains several documents where "area" has the "subject of study" sense. Without disambiguation, we obtained several false positive documents corresponding to "subject of study" or containing falsely related terms, such as "topic", "issue" and "subject".

Finally, we deepened the effectiveness analysis by considering actual text documents (QT1-QT4) for which to find related documents in the collection, as in the "From Methodology" scenario. Such queries contain many terms and can possibly produce a very large number of results: thus, it is essential to evaluate also the induced ranking, so to assess whether the best suggestions are returned in the top positions. Figure 4 shows (on the left) the precision values obtained for QT1 (other queries performed very similarly) at different recall levels, i.e., when a given percentage of relevant documents have been found, and the distance from the ideal ranking (right). Our semantic method is compared to the non-WSD baseline: differently from the latter, it is able to filter the wrong senses and to identify the most significant terms, without being misled by non-relevant ones.

In conclusion, we can observe that, independently from the ambiguity of query terms, our semantic method leads to improvements in precision without compromising recall. Moreover, it gives also a key contribution in retrieving the most relevant results as first in the ranking.

4 Concluding Remarks

Several literature papers have highlighted possible benefits of combined knowledge engineering (KE) and software engineering methods for specific SE tasks [10, 7]. For instance, while standard reuse repositories are limited to plain syntactic search and, thus, generally suffer from low precision and recall [7], knowledge-based methods, such as [8] enhance the effectiveness of the component reuse task by proposing the usage of formal descriptions of components (in OWL) to

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

be queried by specific graph query languages, such as SPARQL. Other notable proposals have been presented, for instance, for facilitating software process assessment through formal descriptions of specific improvement methods, such as CMMI [10]. As already noted in [7], however, the discussion on integrating SE and KE methods has been, in many cases, very academic, focusing on aspects like meta-modeling and neglecting applicability and usability.

Instead, the search method we presented in this paper is designed to be effective and easily applicable. Future work, in the upcoming project evaluation phase, will involve user IT companies in actual scenarios in order to obtain opinions about the effectiveness of the proposed techniques. Moreover, we will study how to adapt our search methodology in the agro-food domain in the context of the Biogest-Siteia projects [4], funded by Emilia-Romagna (Italy) regional government, which aims to increase the competitiveness of Regional seed companies through the use of modern selection technologies.

References

1. DIS 9001:2000 Quality Management Systems - Requirement. In *ISO TC176*, 1999.
2. Carnegie Mellon University Software Engineering Institute. In *CMMI for Development, Version 1.2*, 2006.
3. OSES Architecture and Component Specification. In G. Benguria, editor, *FP7-SME FACIT-SME (FP7-243695), Deliverable*, December 2010.
4. ORM Architecture and Engineering Models. In F. W. Jaekel, editor, *FP7-SME FACIT-SME (FP7-243695), Deliverable*, October 2010.
5. R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
6. S. Banerjee and T. Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *IJCAI*, pages 805–810, 2003.
7. H.-j. Happel and S. Seedorf. Applications of ontologies in software engineering. *Engineering*, pages 1–14, 2006.
8. C. Kiefer, A. Bernstein, and J. Tappolet. Mining Software Repositories with iSPAROL and a Software Evolution Ontology. In *MSR*, page 10. IEEE Computer Society, 2007.
9. C. Leacock and M. Chodorow. *Combining Local Context and WordNet Similarity for Word Sense Identification*, chapter 11, pages 265–283. The MIT Press, 1998.
10. H. Leung, L. Liao, and Y. Qu. A software process ontology and its application. In *the 4th International Semantic Web Conference*, 2005.
11. F. Mandreoli and R. Martoglia. Knowledge-based sense disambiguation (almost) for all structures. *Inf. Syst.*, 36(2):406–430, 2011.
12. R. Martoglia. Facilitate IT-Providing SMEs in Software Development: a Semantic Helper for Filtering and Searching Knowledge. In *SEKE*, pages 130–136. Knowledge Systems Institute Graduate School, 2011.
13. R. Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2), 2009.
14. L. Po and S. Sorrentino. Automatic generation of probabilistic relationships for improving schema matching. *Inf. Syst.*, 36(2):192–208, 2011.
15. G. Salton and C. Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[4] http://www.biogest-siteia.unimore.it/

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

An Efficient Methodology for the Identification of Multiple Music Works within a Single Query*

Emanuele Di Buccio[1], Nicola Montecchio[1], and Nicola Orio[2]

[1] Department of Information Engineering, University of Padova, Padova IT
[2] Department of Cultural Heritage, University of Padova, Padova IT
{emanuele.dibuccio,nicola.montecchio,nicola.orio}@dei.unipd.it

Abstract. A comprehensive methodology for automatic music identification is presented. The main application of the proposed approach is to provide tools to enrich and validate the descriptors of recordings digitized by a sound archive institution. Experimentation has been carried out on a collection of digitized vinyl discs, although the methodology is not linked to a particular recording carrier. Automatic identification allows a music digital library to retrieve metadata about music works even if the information was incomplete or missing at the time of the acquisition. Results show that the approach is both efficient and effective.

Keywords: Music Identification, Audio Indexing, Cultural Heritage

1 Introduction

In this paper we present a complete methodology for music identification. The main application that motivates our approach is the automatic identification of recordings digitized by sound archives. The experimentation is carried out on a collection of vinyl discs, although the methodology is not linked to a particular carrier and can be readily extended to audio tapes, shellack discs, and so on. Automatic identification allows a music digital library system to retrieve relevant metadata about music works even if this information was incomplete or missing at the time of the digital acquisition. We believe that the availability of systems for the identification of music works will promote the dissemination of music cultural heritage, allowing final users to retrieve and to access individual recordings also when descriptive metadata were not available at the time of acquisition.

A typical approach to automatic music identification is based on the extraction of an *audio fingerprint* from digital recordings. A fingerprint is a compact set of music features that allows for the identification of digital copies even in the presence of noise, distortion, and compression; it can be seen as a content-based signature that summarizes an audio recording [1].

The more general task of music identification differs from audio fingerprint because of the variability of timbre, instrumentation and tempo. Identification of *different versions of the same work* is often based on the extraction of a sequence of audio features

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

that are related to high level characteristics of music works – melody, harmony, rhythm – and their alignment using well-known techniques such as Hidden Markov Models (HMMs) and Dynamic Time Warping (DTW) [2].

Identification methodologies based on alignment are usually computationally intensive, because they require the alignment of a query with every item in the reference collection. Quantization of audio features is proposed in [3] to obtain an index-based matching procedure, aimed at efficient music identification. Results showed that the approach is robust to typical variations due to different interpretation of classical music pieces. Our approach, described in detail in [4], is also based on an index data structure to provide an efficient identification methodology.

2 Identification Methodology

Our approach to cover identification aims primarily at efficiency. The methodology is inspired by the concept of Locality Sensitive Hashing (LSH) [5], which is a general approach to handle high dimensional spaces by using ad-hoc hashing functions to create collisions between vectors that are close in the high dimensional space. LSH has been applied to efficient search in media collections [6].

In the proposed approach, the problem of music identification is addressed making use of a particularly compact representation of the audio content, described in Section 2.1. This representation allows the similarity among recordings to be computed by means of a *bag of features* representation, which allows us to exploit indexing techniques to speed up retrieval: Section 2.2 describes how the resulting audio recording representations are stored in an inverted index and the rationale underlying the identification process. Section 2.3 deals with the identification of multiple recordings within a single "long" query.

2.1 Audio Content Representation

A recording is stored in a digital system as a discrete signal, representing an acoustic pressure measured at regular time intervals. In the proposed approach, this *audio waveform* is divided into short overlapping excerpts of fixed length (dozens of milliseconds), and content-based descriptors are then extracted from each resulting segment. The adopted descriptors are *chroma features* [7], 12-dimensional vectors representing the energy associated to each pitch class in a short time frame. A pitch class is a set of pitches corresponding to a given note in the chromatic scale. For instance, the pitch class of the note C corresponds to frequencies: 32.7 Hz, 64.4 Hz, 130.8 Hz, and so on.

Chroma extraction is preceded by *tuning frequency adjustment* in order to be robust to adoption of different reference frequencies, and is followed by a *key finding* procedure, which allows us to deal with different versions of the same music work performed in different keys. The choice of key invariant chroma features as content descriptors is based on the consideration that listeners use mostly harmonic and melodic cues to decide whether two recordings are different performances of the same music work. Differences in tonality, tempo and duration do not substantially affect the similarity

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

judgment and thus our representation aims at being robust to changes in these music dimensions.

A general formula to compute chroma vectors from a windowed signal $s(t)$ is the following

$$c_i = \sum_{f=32Hz}^{4000Hz} B_i(f) \cdot S(f) \tag{1}$$

where $S(f)$ is a representation of $s(t)$ in the frequency domain and $B_i(f)$ is a bank of bandpass filters, each centered on the semitones belonging to pitch class i and with a bandwidth of a semitone. Usually chroma features are computed only on a limited range of the audible spectrum.

The particular choice of tuning frequency adjustment, key finding, and chroma extraction algorithms is crucial to the identification accuracy; however, since the focus of the paper is on the identification methodology, rather than signal analysis of music content, we refer the reader to [4] for detailed descriptions of the mentioned algorithms.

Once audio recordings have been represented by chroma vectors, a *hashing* step is introduced to allow for a compact representation of chroma vectors. The quantized version q of a chroma vector $\mathbf{c}$ is obtained by taking into account the ranks of the chroma pitch classes, sorted by their values. Let r_i be the position in which the i-th component c_i would be ranked after a descending sort (starting from 0); a k-level rank-representation of $\mathbf{c}$ is constructed by considering a base 12 number computed as:

$$r_i = |\{c_j : c_j > c_i, j = 1, \ldots, 12\}| \qquad i = 1, \ldots, 12 \tag{2}$$

$$q = \sum_{i:r_i<k} i \cdot 12^{r_i} \tag{3}$$

2.2 Efficient Identification

The identification algorithm is based on a two-level bag-of-features representation of the audio content: the hash sequence computed from the audio waveform according to the procedure described above is divided into overlapping segments of fixed length (typically corresponding to dozens of seconds). Identification of an unknown recording is carried out computing its similarity with the recordings in the collection; adopting the textual IR terminology, we will refer to them as *query* and *documents*, respectively.

Let Q (D) denote a recording associated to a query (audio document in the database), composed of a sequence $\{q_1 \ldots q_{|Q|}\}$ ($\{d_1 \ldots d_{|D|}\}$) of hash sequences, each of length $|q|$ ($|d|$). It is possible to conceptually distinguish two phases in the similarity computation procedure. The first step implies the computation of local similarity between segments d and q as the (normalized) number of descriptors they have in common

$$S_L(d, q) = \sum_{t \in q \cap d} \min\Big(\frac{\mathrm{tf}(t, d)}{|d|}, \frac{\mathrm{tf}(t, q)}{|q|}\Big) \tag{4}$$

where $\mathrm{tf}(t, d)$ denotes the count (*term frequency*) of hashes with value t in segment d. The second step aggregates the contributions of all the query segments, by computing

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

the geometric mean of the best local similarity values for each query segment:

$$S(Q, D) = \sqrt[|sQ|]{\prod_{q \in Q} \max_{d \in D} S_L(d, q)} \quad (5)$$

An efficient implementation of the similarity computation procedure is possible because any information regarding the *ordering* of segments, and of the hashes contained therein, is discarded. This is reflected in the notation of Equation 5 by the exclusive use of *set operations*.

Data Representation The computation of the similarity score for a query-document pair requires information on the hash frequency in each query and document segment. Information on the frequency of occurrence of an hash in a specific query segment can be extracted at query time and efficiently accessed by means of data structure maintained in memory. The current implementation of the architecture exploits a list of maps, where each list entry (each map) corresponds to a segment and retains *(hash,frequency)* pairs.

Indexing Information on the frequency of a descriptor in a document can be efficiently accessed by means of an inverted index. In such data structure an *inverted list* is associated to each distinct descriptor appearing in at least one of the documents in the collection; the entries of the inverted list are the (identifiers of the) documents where the descriptor occurs. Moreover, additional information necessary by the ranking function can be stored in the inverted list entries, e.g. the frequency of occurrence of the descriptors or the positions where they occur.

In the adopted methodology, the descriptors are chroma hashes, and each segment of a recording is interpreted as a textual document. Figure 1 provides an example of the indexing process when applied to a recording, represented as a sequence of hashes. Such sequence undergoes a segmentation process where possibly overlapping subsequences are extracted from the recording sequence. After segmentation, a recording is represented by a *set* of hash sequences. In accordance with the *bag of features* paradigm, it is hypothesized that information on hash occurrences at a segment level is sufficient for effectively identify a recording; positional information is therefore ignored, thus each segment is effectively treated as a *set* of hashes. An inverted index can efficiently store hash occurrence information: the list of index descriptors is the set of distinct hashes in all the recordings in the collection, while the entries in the inverted list for a hash are the *(segment,frequency)* pairs for that hash.

Document At A Time Processing The adoption of an inverted index allows us to compute efficiently the segment similarity score in Equation 4. When a query Q is submitted to the system it undergoes the same steps as the documents in the collection. The key finding algorithm mentioned in Section 2.1 is applied to the query, thus obtaining diverse transpositions that the system treats as distinct queries, processed in parallel. After the computation of its hash values, the query is split into a number of *segment-queries*. Each segment-query is processed by a Document At A Time (DAAT) strategy. DAAT strategies evaluate the contributions of every query term with respect to a single document before considering the next document. One advantage of the DAAT strategy is

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

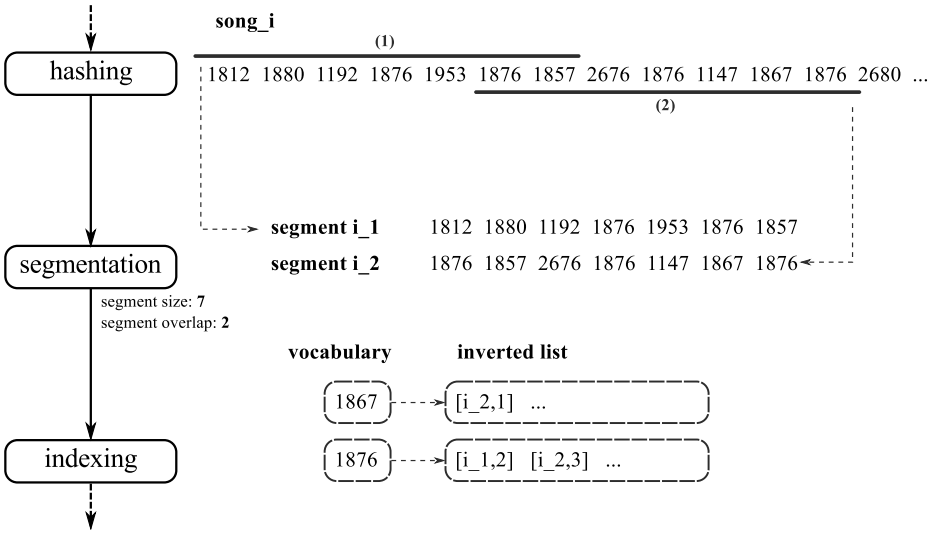


Fig. 1. Segmentation and indexing of hashed chroma descriptors.

that it does not require intermediate document scores to be maintained during the entire ranking process, thus limiting the run-time memory usage.

Score Fusion For each segment-query the system returns a ranked list whose entries are the document segments of all the documents in the collection ranked in decreasing order of similarity value computed at the segment level. The maximum among all the document segments is then computed by going through the returned ranked list. The next step consists in the aggregation of the results returned by the segment-queries, according to Equation 5. Finally, the results obtained from the diverse considered transpositions are merged to obtain a final results list. An advantage of this retrieval strategy is that the diverse query (transpositions) and the diverse segment-queries can be processed in parallel.

2.3 Identification of Multiple Works Within a Single Query

The algorithm detailed above assumes that a query can be matched with a recording of the same music material. This is the typical scenario for music identification systems, but in order to handle the case of a query containing multiple works – such as digitizations of tapes or vinyl discs containing several tracks – the methodology must be extended.

To this end, an additional time-resolution level was added to the segmentation hierarchy. Basically, the recording of an LP side is divided into shorter elements, each one considered as an individual query. Figure 2 shows the procedure, which provides us with a number of resulting rank lists that are merged into a structure called *rank ma-*

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

trix. We will refer to these short elements as "chunks", to disambiguate them from the segments in which a single track is divided.

The rationale behind this choice is related to the way the similarity between a query and the documents is computed. As can be seen from Equation 5, it involves a maximization over indexed segments of the local similarities for each query segment. As a consequence, the system is designed to support short queries containing a portion of a corresponding recording while it becomes ineffective with long queries containing segments of different recordings, because the geometric mean will be computed also from local scores with very low similarity values. It should be noticed that, while the maximization and averaging indexes of Equation 5 – d and q respectively – could be in principle reversed (there is no theoretical reason for the asymmetry of the global similarity function), several implementation choices aimed at efficiency depend on this configuration.

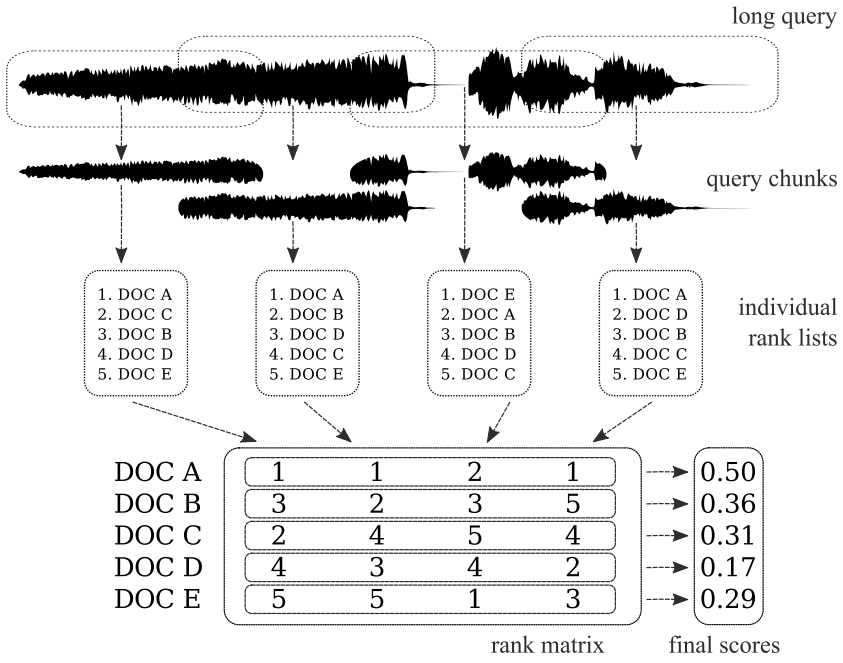


Fig. 2. Computation of the rank matrix for chunks of a long query.

The similarity S_i, between the query and the recording indexed in the i-th row of the rank matrix, is computed with a simple data fusion approach:

$$S_i = \max_{w=1\ldots L-W} \sum_{j=w}^{w+W-1} \frac{1}{(\max(r_{i,j}, C))^2} \tag{6}$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

where L denotes the number of query chunks (columns of the rank matrix), and $r_{i,j}$ represents the position of recording i in the rank list produced from chunk j. The underlying idea is to consider each row of the rank matrix, and to analyze small windows of length W: an indexed recording spanning multiple chunks is supposed to consistently rank in high positions, and the windowing (along with the saturation constant C) acts as a filter for documents whose behavior in the ranking sequence is noisy.

The choice of using rank values instead of similarity scores is motivated by the consideration that averaging similarity scores works only if all segments belong to the same recording, whereas different recordings induce radically different similarity values. As for many data fusion approaches, the choice of rank values reduces the effect of large differences in the similarity scores. Moreover, the choice of ranks enables independence of the particular parametrization adopted for the local similarity computation.

3 Experimental Evaluation

The objective of the experiments reported in this section is to evaluate the capability of the proposed methodology to identify multiple works within a music recording. In particular, it should be verified whether a fixed-length segmentation strategy is able to support identification of recordings characterized by radically different durations.

3.1 Test Collection

The adopted experimental collection is composed of two corpora. The first is part of the test collection that was recently introduced and adopted in the Music Identification Task of the MusiCLEF Lab in CLEF2011 [8]. It is made up of 6680 tracks, grabbed from commercial CDs; among those, 2671 music works are represented at least twice in the collection, forming 945 cover sets. The second corpus is a collection provided by the Fonoteca of the University of Alicante and constituted of 100 LPs that were digitized using common LP record player equipment. The LPs from the Fonoteca were used to evaluate the procedure detailed in Section 2.3, matching their contents with recordings in the MusiCLEF2011 collection.

3.2 Results

The accuracy of the identification methodology is greatly affected by the choice of a particular combination of parameters, among which are the chroma extraction and the tuning frequency adjustment algorithms, and the values of the quantization and segmentation parameters. In order to select an effective combination of parameters, we used the reference recording dataset of the MusiCLEF2011 collection. Considering the accuracy/efficiency trade off, we selected the chroma feature extraction algorithm proposed in [9], using segments of 50s without overlap and 3-level chroma quantization. The results obtained for all the different configurations are reported and discussed in [4].

The chosen parametrization was subsequently used to perform the LP identification task, where multiple works are present within a single query. The effect of different

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Table 1. Experimental results, on a database of 6680 recordings, using a 3.4 Ghz machine.

Task	MRR	MAP	mean query time	mean query length
Track identification	.832	.766	2s	5'25"
LP identification	.900	.742	46s	47'13"

choices for the segmentation and length of the analysis window on accuracy and efficiency of the algorithm was also investigated.

Table 1 reports the accuracy of the algorithm in terms of Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP), along with the average query time (in the LP identification case, the average query time refers to the time needed to identify a complete LP). The experiments were performed on a machine featuring a dual-core 3.4 Ghz CPU and a 7200 RPM hard disk.

The proposed approach is both effective and efficient. In particular, identification can be carried out in a small fraction of the time required for ripping a track from a commercial CD and, most of all, digitizing a complete LP.

References

1. Cano, P., Batlle, E., Kalker, T., Haitsma, J.: A review of audio fingerprinting. Journal of VLSI Signal Processing **41** (2005) 271–284
2. Serrà, J.: Identification of versions of the same musical composition by processing audio descriptions. PhD thesis, Universitat Pompeu Fabra, Barcelona (2011)
3. Kurth, F., Müller, M.: Efficient index-based audio matching. IEEE Transactions on Audio, Speech, and Language Processing **16**(2) (2008) 382–395
4. Montecchio, N., Di Buccio, E., Orio, N.: An efficient identification methodology for improved access to music heritage collections. Journal of Multimedia **7**(2) (2012) 145–158
5. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: Proceedings of the International Conference on Very Large Data Bases, Edinburgh, UK (1999)
6. Slaney, M., Casey, M.: Locality-sensitive hashing for finding nearest neighbors [lecture notes]. Signal Processing Magazine, IEEE **25**(2) (2008) 128–131
7. Fujishima, T.: Realtime chord recognition of musical sound: a system using common lisp music. In: Proceedings of the International Computer Music Conference, Beijing, China (1999)
8. Orio, N., Miotto, R., Montecchio, N., Rizo, D., Lartillot, O., Schedl, M.: Musiclef: a benchmark activity in multimodal music information retrieval. In: Proceedings of the Internation Society for Music Information Retrieval conference, Miami, FL, USA (2011)
9. Lartillot, O.: A comprehensive and modular framework for audio content extraction, aimed at research, pedagogy, and digital library management. In: Proceedings of the Audio Engineering Society convention, London, UK (2011)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Using Keywords to Find the Right Path through Relational Data*

Roberto De Virgilio, Antonio Maccioni, and Riccardo Torlone

Università Roma Tre, Italy
{dvr,maccioni,torlone}@dia.uniroma3.it

Abstract. Keyword search is emerging as the standard way to access information of any kind. Following this trend, several approaches to keyword search over relational databases have been proposed recently. Most of them build tree-shaped solutions that connect tuples matching the given keywords. In this paper, we propose a novel technique to this problem that generates solutions by simply combining joining paths. In this way, the complexity of the process is reduced and the query engine of the underlying RDBMS can be fully exploited. Several experiments show a very good behavior with respect to other approaches in terms of both effectiveness and efficiency.

1 Introduction

With the size and availability of data constantly increasing, it is usually hard for users to retrieve the information they really need. In a frequent scenario, the problem is caused by the fact that the search is over structured data stored in database systems. Then, the users must be aware of the schema and know the syntax of a specific query language. For this reason, alternatives ways to access information are increasingly capturing the attention of database researchers. Among them, several approaches to keyword-based search over structured and semi-structured data have been proposed recently. Typically, keyword-based search over relational data involve the following steps: (i) selection of the tuples whose values match the input keywords, (ii) generation of tree-shaped solutions built by joining the retrieved tuples, and (iii) ranking of the solutions according to a relevance criteria. At the end, only the top-K solutions are usually returned.

In this paper, we present a novel technique for keyword-based search over relational databases that builds the best results using a path-oriented strategy. This process is inspired by an earlier work on keyword search over semantic data [3]. Unlike many current approaches that do not exploit enough the capabilities of the underlying system [9], our technique retrieves efficiently the tuples that match the given keywords by taking full advantage of the RDBMS in which the data are stored. We have tested the approach over available benchmarks and have observed a very good behavior with respect to other proposals in terms of both effectiveness and efficiency.

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

The most prominent work in this area can be classified in *schema-based* and *schema-free* approaches. Schema-based approaches [6,8,9] are common for relational databases. Here, an answer to a query is a tree, usually called *joined tuples tree* (JTT), composed by joining tuples. A common drawback of these approaches is that the search is converted into a set of SQL statements that generate candidate answers. All queries are then executed but some of them can return empty results, leading to inefficiency, which is likely to worse with the size of the data set. Schema-free approaches (*e.g.* [7]) are more general as they operate on arbitrary graph-shaped data. A general approach searches for the top-K connected trees, each representing a *(minimal) Steiner tree* [4]. A relevant drawback is that finding a minimal Steiner tree is an NP-Hard problem [4]. Hence this methods rely on complex heuristics aimed at generating approximations of Steiner trees. Our approach tries to overcome these limitations, trading-off accuracy in top-K computation, and scaling seamlessly with the size of the input.

In the rest of the paper, we first introduce, in Section 2, a path-oriented model for relational databases. Based on this model, in Section 3 we present our technique for answering keyword search queries and illustrate, in Section 4, some experimental results.

2 Path-oriented Modeling

A relational database $\mathcal{RDB}$ can be modeled by a pair of graphs $\langle \mathcal{SG}, \mathcal{DG} \rangle$ representing the schema and the instance of $\mathcal{RDB}$, respectively, as follows.

Definition 1 (Schema Graph) *Given a relational schema* $\mathcal{RDB}$-SC $= \langle \mathcal{T}, \mathcal{A} \rangle$, *where* $\mathcal{T}$ *is a set of relation schemas* T_i *over a set of attributes* $\{T_i.A_1, \ldots, T_i.A_k\}$ *and* $\mathcal{A}$ *is the union of all attributes, a schema graph* $\mathcal{SG}$ *on* $\mathcal{RDB}$-SC *is a directed graph* $\langle V, E \rangle$ *where* $V \subseteq \{\mathcal{T} \cup \mathcal{A}\}$, $E \subseteq \{(\mathcal{T} \times \mathcal{A}) \cup (\mathcal{A} \times \mathcal{A})\}$ *and there is an edge* $(a, b) \in E$ *if (i)* $a \in T$ *and* b *is an attribute of* a, *(ii)* $a \in \mathcal{A}$ *belongs to a key of* T_i *and* b *is an attribute of* T_i, *and (iii) there is a foreign key between* a *and* b.

T_1: **Person**

	name	area
t_1	Watson	Database
t_2	Lenzerini	Database
t_3	Date	Database
t_4	Hunt	Inf. Systems

T_2: **Affiliated**

	professor	department
t_5	Watson	x123
t_6	Lenzerini	cs34
t_7	Date	cs34
t_8	Hunt	m111

T_3: **Department**

	id	dname	director
t_9	x123	CS	Watson
t_{10}	cs34	IE	Hunt
t_{11}	ee67	EE	Date
t_{12}	m111	ME	Hunt

T_4: **Author**

	name	publication
t_{13}	Lenzerini	Data Integration
t_{14}	Date	Foundation Matters

T_5: **Publication**

	title	year
t_{15}	Data Integration	2002
t_{16}	Foundation Matters	2002

Fig. 1. A relational database

For instance, in the relational database $\mathcal{RDB}$-$SC_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ depicted in Fig. 1 [1] we have: $\mathcal{T}_1 = \{$*Person, Affiliated, Department, Author, Publication*$\}$, and $\mathcal{A}_1 = \{$*Person.name, Person.area, Affiliated.professor*, $\ldots\}$. In the figure, underlined attributes are primary keys and we have the following foreign key constraints: *Affiliated.professor* $\xrightarrow{fk}$ *Person.name*, *Affiliated.department* $\xrightarrow{fk}$ *Department.id*, *Department.director* $\xrightarrow{fk}$ *Person.name*, *Author.name* $\xrightarrow{fk}$ *Person.name*, and

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Author.publication $\xrightarrow{fk}$ *Publication.title*. The schema graph $\mathcal{SG}_1$ on $\mathcal{RDB}$-SC_1 is depicted in Fig. 2. In a schema graph the sources represent the relations of a

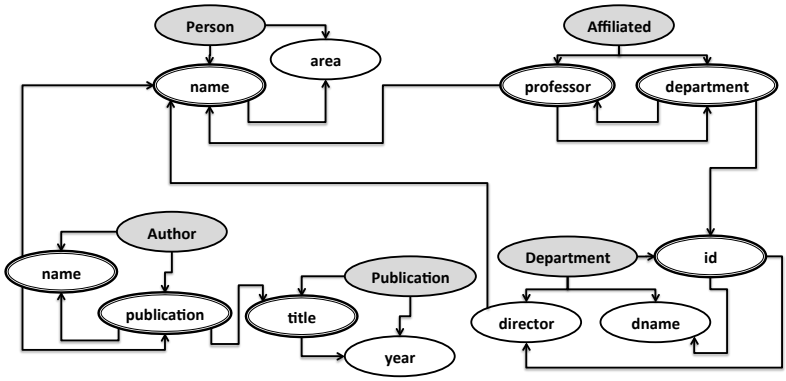


Fig. 2. The schema graph for the relational database in Fig 1.

relational database schema (grey vertices) and the so-called *Schema-Paths* track the relationships between attributes, according to primary and foreign keys.

Definition 2 (Schema-Path) *Given a schema graph $\mathcal{SG} = \{V, E\}$, a schema-path is a sequence $sp = v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_f$ where $(v_i, v_{i+1}) \in E$ and v_1 is a relation node.*

In Fig. 2 a schema-path *sp* is *Affiliated* → *Affiliated.professor* → *Person.name*. We introduce an injective function called *index*, denoted by idx, that maps each tuple t of a relation for a schema $T \in \mathcal{T}$ to a *tuple-id* (tid for short) belonging to an uncountable set $\mathbb{U}$.

Definition 3 (Data Graph) *Given a relational database instance $\mathcal{RDB}$-I $= \langle \mathcal{T}, \mathcal{A}, I, \mathcal{D} \rangle$, where I ($\mathcal{D}$) is the set of all tids (values) occurring in the database, a data graph $\mathcal{DG}$ on $\mathcal{RDB}$-I is a directed graph $\langle V, E \rangle$ where $V \subseteq \{\mathcal{T} \cup \mathcal{A} \cup I \cup \mathcal{D}\}$, $E \subset \{(\mathcal{T} \times \mathcal{A}) \cup (\mathcal{A} \times I) \cup (I \times \mathcal{D})\}$ and there is an edge $(v_1, v_2) \in E$ if: (i) $v_1 \in \mathcal{T}$ and v_2 is an attribute of v_1, (ii) $v_1 \in \mathcal{A}$ belongs to a key of T_i and v_2 is the tid of a tuple for T_i, and (iii) v_1 is a tid and v_2 is a value of a tuple t such that $v_1 = \text{idx}(t)$.*

Fig. 3 shows an example of data graph for the database of Fig. 1.

Definition 4 (Data-Path) *Given a data graph $\mathcal{DG} = \{V, E\}$, a data-path is a sequence $dp = v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$ where $v_i \in V$, $(v_i, v_{i+1}) \in E$, v_1 is a relation node, v_2 is an attribute node, v_3 is a tuple-id and v_4 is a data value.*

In Fig. 3 a data-path dp_k is *Person* → *Person.name* → t_1 → *Watson*. A data-path $dp = T_i \rightarrow T_i.A_j \rightarrow \text{idx}(T_i.tp_s) \rightarrow dv$ can be *expanded* using a schema-path *sp* as follows.

Definition 5 (Expanded Data-Path) *Given a schema-path $sp = T_1 \rightarrow T_1.A_1 \rightarrow T_2.A_i \rightarrow T_3.A_j \rightarrow \ldots \rightarrow T_n.A_z$ and a data-path $dp = T_n \rightarrow T_n.A_z \rightarrow \text{idx}(T_n.t_s) \rightarrow v$, the expanded data-path dp' of dp through sp is the path $T_1 \rightarrow T_1.A_1 \rightarrow ?x_1 \rightarrow T_2.A_i \rightarrow ?x_2 \rightarrow T_3.A_j \rightarrow ?x_3 \rightarrow \ldots \rightarrow ?x_{n-1} \rightarrow T_n.A_z \rightarrow \text{idx}(T_n.t_s) \rightarrow v$, where each $?x_i$ is a variable ranging over the set of tids.*

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

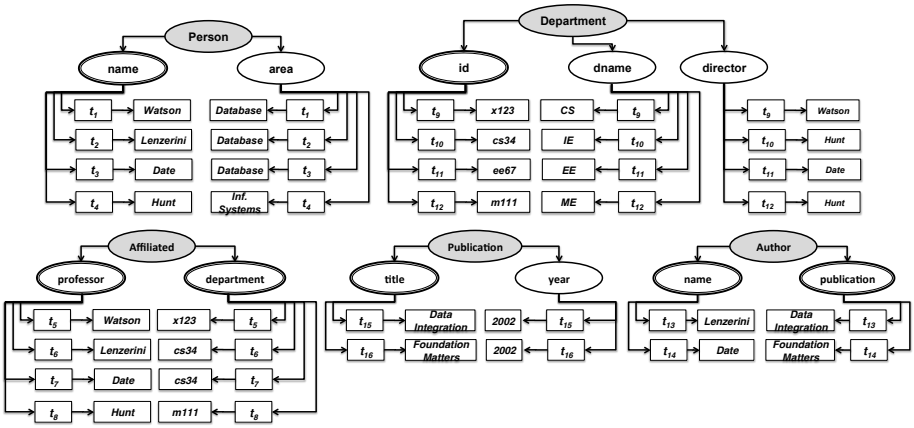


Fig. 3. The data graph for the relational database in Fig 1.

Let us consider again the example in Fig. 3. The data-path dp = *Publication* → *Publication.title* → t_{16} → *Foundation Matters (F.M.)* can be expanded with the schema-path sp = *Author* → *Author.name* → *Author.publication* → *Publication.title* as follows:

$$Author \rightarrow Author.name \rightarrow ?x_1 \rightarrow Author.publication \rightarrow ?x_2 \rightarrow Publication.title \rightarrow t_{16} \rightarrow F.\ M.$$

This path describes the fact that the tuple with tid t_{16} has a relationship with the attribute *publication* of a tuple ($?x_2$) in the relation *Author*. Moreover, the same tuple ($?x_2$) is related to the *name* value of tuple ($?x_1$) of the same relation *Author*; for transitivity $?x_1$ is related to t_{16}, too. Referring to Fig. 3, both $?x_1$ and $?x_2$ correspond to the tuple with tid t_{14}. The path specifies that *Date* is the author of *Foundation Matters*, a publication of *2002*. Two (expanded) data-paths are correlated if they have a common node.

We adopt an Information Retrieval approach for matching values that rely on traditional full text search. In the rest of the paper we will denote the matching relationship with ≈. Intuitively, a solution to a keyword-based query Q is a set of tuples such that: (i) they match the keywords in Q and (ii) they occur in a set of expanded data-paths that, taken together, form a connected graph. More formally, a solution can be defined as follows.

Definition 6 (Solution) *Given a query Q made of a set of keywords $\{z_1, \ldots, z_n\}$, a solution S of Q is a set of expanded data-paths DP such that: (i) for each $z_i \in Q$ there exists a tuple t in DP that matches z_i* (i.e. $t \approx z_i$)*, and (ii) for each pair of expanded data-paths $dp_a, dp_b \in DP$ there is a sequence of expanded data-paths $[dp_a, dp_i, \ldots, dp_j, dp_b]$ occurring in DP such that each element of the sequence has at least a node in common with the following.*

Clearly, given a solution to a query, the answer to the final user is just the set of all tuples whose tids are contained in a solution.

Intuitively, a solution S_1 is more relevant than another solution S_2 if S_1 is more compact than S_2 since, in this case, the keywords of the query are closer between each other in S_1. This is captured by a scoring function that defines the relevance of solutions, as follows.

The size of a solution S, denoted by $\lambda(S)$, is the sum of the length of all its expanded data paths.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Definition 7 (Ranking of solutions) *Given two solutions S_1 and S_2 of a query Q we say that S_1 is more relevant than S_2 if $\lambda(S_1) \geq \lambda(S_2)$. The ranking of a set of solutions is a function ρ such that $\rho(S_1) \geq \rho(S_2)$ if and only if S_1 is more relevant than S_2.*

Problem Statement. Given a relational database $\mathcal{RDB}$ and a keyword search based query $Q = \{z_1, z_2, \ldots, z_n\}$, where each z_i is a keyword, we aim at finding the top-K ranked solutions $S_1, S_2, \ldots, S_k$.

For example, the best solution for the query $Q_1 = \{\texttt{Matters}, \texttt{Database}\}$ is $S_1 = \{t_3, t_{14}, t_{16}\}$ as will be better clarified in the next section.

3 Path-oriented Search

Once the relational database $\mathcal{RDB} = \langle \mathcal{SG}, \mathcal{DG} \rangle$ is indexed in order to gain immediate access to the information of interest (as detailed in Section 4), the query evaluation takes place. First of all, the paths are organized in clusters.

Clustering. There are as many clusters as the number of keywords in the query Q; for each $z_i \in Q$ we retrieve all data paths dp from $\mathcal{DG}$ such that dp ends into a data value v matching z_i. Then we expand dp with each schema path $sp \in \mathcal{SG}$ ending into the attribute node of dp. Finally we insert dp and each resulting expanded data-path into the corresponding cluster. Each cluster is implemented as a priority queue where the priority decreases with the increasing length of a path. Referring to the query $Q_1 = \{\texttt{Matters}, \texttt{Database}\}$ over the relational database in Fig. 1, we obtain two clusters

$$cl_{Matters} : \begin{pmatrix} dp'_1 : \texttt{Author} \rightarrow \texttt{Author.publication} \rightarrow \texttt{t}_{14} \rightarrow F.M. \\ dp'_2 : \texttt{Publication} \rightarrow \texttt{Publication.title.} \rightarrow \texttt{t}_{16} \rightarrow F.M. \\ dp'_3 : \texttt{Author} \rightarrow \texttt{Author.name} \rightarrow ?x_1 \rightarrow \texttt{Author.publication} \rightarrow \texttt{t}_{14} \rightarrow F.M. \\ dp'_4 : \texttt{Author} \rightarrow \texttt{Author.publication} \rightarrow ?x_2 \rightarrow \texttt{Publication.title} \rightarrow \texttt{t}_{16} \rightarrow F.M. \\ \ldots \end{pmatrix}$$

$$cl_{Database} : \begin{pmatrix} dp'_5 : \texttt{Person} \rightarrow \texttt{Person.area} \rightarrow \texttt{t}_1 \rightarrow Db \\ dp'_6 : \texttt{Person} \rightarrow \texttt{Person.area} \rightarrow \texttt{t}_2 \rightarrow Db \\ dp'_7 : \texttt{Person} \rightarrow \texttt{Person.area} \rightarrow \texttt{t}_3 \rightarrow Db \\ \ldots \\ dp'_8 : \texttt{Author} \rightarrow \texttt{Author.name} \rightarrow ?x_3 \rightarrow \texttt{Person.name} \rightarrow ?x_4 \rightarrow \texttt{Person.area} \rightarrow \texttt{t}_3 \rightarrow Db \\ \ldots \end{pmatrix}$$

Building. Once we have computed the expanded data-paths organized in $\mathcal{CL}$, we combine them to provide the best k solutions to Q. The building algorithm (Alg. 1) is an incremental process. In the following we refer to data-paths including also the expanded paths. Every step starts dequeuing the best paths (*i.e.* the shortest ones) from each cluster into a set DP (`dequeueTop`). Then, the procedure `combinations` generates all the possible combinations of paths within DP in order to find all connected graphs representing candidates of solutions. This is done by taking exactly one data-path from each cluster. For instance referring to our example, at the first running of the algorithm we have to combine dp'_1, dp'_2 from $cl_{Matters}$ with dp'_5, dp'_6, dp'_7 from $cl_{Database}$; we obtain six pairs, *e.g.* $\{dp'_1, dp'_5\}$, $\{dp'_1, dp'_6\}$ and so on. A combination c represents a solution if and only if there exists a *center* in c, that is, a tuple(-id) from which it is possible to reach the tuples matching all the keywords in Q. This evaluation is performed by the procedure `backward_exploration`.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

Algorithm 1: Building

Input : The set of clusters $\mathcal{CL}$, a query Q, the number k.
Output: The set of solutions $\mathcal{S}$.

```
1  finished ← false;
2  S ← ∅;
3  while ¬finished do
4      DP ← ∅;
5      foreach cl_i ∈ CL do
6          DP ← DP ∪ dequeueTop(cl_i);
7      if CL = ∅ then finished ← true;
8      else
9          // we re-enqueue last extracted dp in the corresponding
             empty cl_i for combining such dp with non empty clusters'
             data-paths
10         foreach cl_i ∈ CL: cl_i = ∅ do
11             foreach dp ∈ DP: dp.vf = cl_i.keyword do
12                 cl_i.enqueue(dp);
13     C ← combinations(DP );
14     foreach c ∈ C do
15         P ← ∅;
16         foreach dp ∈ c do
17             is_sol ← backward_exploration(dp, Q, P);
18         if is_sol then
19             S.enqueue(P.keys) ;      // The priority is score(P.keys)
20             if |S| = k then finished ← true;
21 return S;
```

Intuitively the best solution contains tuples strictly correlated, *e.g.* tuples of the same table. In our representation such kind of solution corresponds to a set of data-paths with the shortest length, *i.e.* a connected graph with the smallest diameter. The building process follows this direction: first of all we try to compose the shortest paths from each cluster, that are good candidates to be the best solutions. If we didn't find k solutions, we try with longer data-paths that would bring to new tuples and potentially to new interconnections and other solutions. Of course, the solutions generated later will present lower score, since the diameter of the corresponding graph of data-paths increases. The combinations produced in the first running of the building are discharged since the pairs of data-paths do not present any intersection. Selecting longer data-paths, we obtain a valid combination $c' = \{dp'_8, dp'_4\}$. Now we have to verify if c' is a solution to include in the set $\mathcal{S}$. Such evaluation explores a data-path in backward: in other terms we follow the foreign key constraints contrariwise. We have two cases: (i) two data-paths correspond to the same table or (ii) two data-paths correspond to two tables linked by a foreign key constraint. In the former case, we instantiate the variables of extended data-paths by extracting the data value associated to the attribute a of the same tuple (*i.e.* it is a simple projection $\pi_a(\sigma_{tid=t}(T))$). In the latter case we follow the key constraint and we need to extract from the new relation T the tuple having the data value v associated to the attribute a (*i.e.* it is a simple selection $\sigma_{a=v}(T)$). The algorithm ends when k solutions are found or when $\mathcal{CL}$ is empty (i.e. we have no more data-paths to combine). We remark that in the building process we exploit operations of

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

selection (σ) and *projection* (π) on limited groups of tuples (often only one) and we never utilize join ($\bowtie$) operations. Let us consider the exploration of the

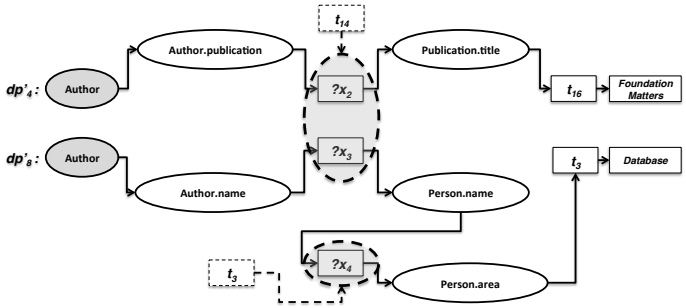


Fig. 4. The building process

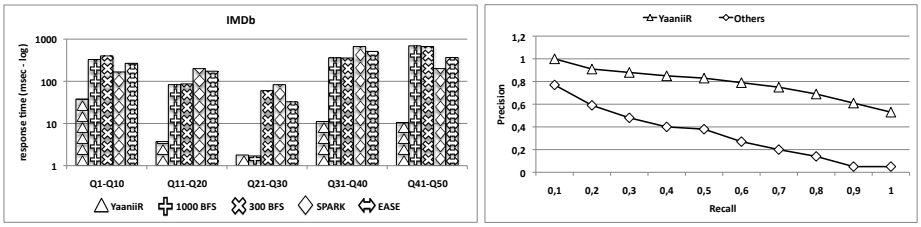
combination $\{dp'_8, dp'_4\}$ of our example, as shown in Fig. 4. In this case the backward exploration starts from analyzing the variable $?x_4$ in the data-path dp'_8. In this case we have a simple projection $\pi_{name}(\sigma_{tid=t_3}(Person))$ to assign the value t_3 to $?x_4$. Following the result of this projection (*i.e.* the value *Date*) allows to process the selection $\sigma_{name='Date'}(Author)$ that assigns the value t_{14} to $?x_3$. Similarly in dp'_4 the selection $\sigma_{publication='Foundation\ Matters'}(Author)$ assigns the same value t_{14} to $?x_2$. Since both $?x_2$ and $?x_3$ present the same value, they represent a node in common between dp'_4 and dp'_8; this means that t_{14} is the center of the graph $\{dp'_8, dp'_4\}$, able to reach all the tuples matching the keywords of the query (*i.e.* t_3 and t_{16}). In this case from the data-paths we extract all tuple-ids (*i.e.* t_3, t_{14}, t_{16}) that represents the first solution to provide.

4 Experimental Results

We implemented our approach in YaaniiR, a Java system for keyword search over relational databases. In our experiments we used the benchmark provided by Coffman et al. [2]. We employ three datasets, IMDb, Wikipedia, and Mondial. For each dataset, we run the set of 50 queries provided in [2]. Experiments were conducted on a dual core 2.66GHz Intel Xeon, running Linux RedHat, with 4 GB of memory, and we used PostgreSQL 9.1 as RDBMS.

Performance. Our algorithms have been implemented in terms of PL/pgSQL procedures. Such procedures exploit a simple index based on two tables: SG(*attribute,path*) and DG(*value,path*). The former stores all schema-paths while the latter all data-paths. Such index is built efficientlty: from few milliseconds on Mondial to a couple of minutes on IMDb and Wikipedia. We compared YaaniiR with the most related approaches: SPARK [8], EASE [7], and the best performing techniques based on graph indexing, i.e. 1000 BFS and 300 BFS that are two configurations of Blinks [5]. For each dataset, we group the queries in five sets: each set is homogeneous with respect to the complexity of the queries (e.g. number of keywords, number of results and so on). For instance referring to IMDb, the first set (i.e. Q1-Q10) searches information about the actors (providing the name as input). The other sets combine actors, movie and characters. For

each set, we ran the queries ten times and measured the average response time (i.e. for computing the top-10 answers). The query response times are shown in Fig. 5 (in *ms* and logarithmic scale). Due to space constraints, in the figure. we report times only on IMDb and Wikipedia, since their much larger size poses more challenges. However the performance on Mondial follows a similar trend. In general Spark and EASE are comparable with Blinks. Our system per-



(a) (b)

Fig. 5. (a) Response Times on IMDb and (b) Precision-Recall interpolation

forms consistently better (in any strategy) for most of the queries, significantly outperforming the others in some cases (e.g. sets Q21-Q30 or Q31-Q40). This is due to the greatly reduced (time) complexity of the overall process with respect to the others spending much time traversing large amount of tuples.

Effectiveness. We have also evaluated the effectiveness of results. We measured the interpolation between precision and recall to find the top-10 solutions, on the queries on all datasets. We calculate the top-10 interpolated precision curve averaged over the systems: Fig. 5.(b) shows the results. As to be expected, the precision of the other systems dramatically decreases for large values of recall. On the contrary our strategies keeps values on the range [0.4,0.8].

References

1. Bergamaschi, S., Domnori, E., Guerra, F., Lado, R.T., Velegrakis, Y.: Keyword search over rel. databases: a metadata approach. In: SIGMOD. pp. 565–576 (2011)
2. Coffman, J., Weaver, A.C.: A framework for evaluating database keyword search strategies. In: CIKM. pp. 729–738 (2010)
3. De Virgilio, R., Cappellari, P., Miscione, M.: Cluster-based exploration for effective keyword search over semantic datasets. In: ER. pp. 205–218 (2009)
4. Garey, M.R., Graham, R.L., Johnson, D.S.: The complexity of computing Steiner minimal trees. SIAM Journal on Applied Mathematics 32(4), 835–859 (1977)
5. He, H., Wang, H., Yang, J., Yu, P.S.: Blinks: ranked keyword searches on graphs. In: SIGMOD. pp. 305–316 (2007)
6. Hristidis, V., Gravano, L., Papakonstantinou, Y.: Efficient ir-style keyword search over relational databases. In: VLDB. pp. 850–861 (2003)
7. Li, G., et al.: Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In: SIGMOD. pp. 903–914 (2008)
8. Luo, Y., Lin, X., Wang, W., Zhou, X.: Spark: top-k keyword query in relational databases. In: SIGMOD. pp. 115–126 (2007)
9. Qin, L., Yu, J.X., Chang, L.: Keyword search in databases: the power of rdbms. In: SIGMOD. pp. 681–694 (2009)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Efficient Diversification of Top-k Queries over Bounded Regions*

Piero Fraternali, Davide Martinenghi, and Marco Tagliasacchi

Dipartimento di Elettronica e Informazione – Politecnico di Milano
Piazza Leonardo da Vinci, 32 – 20133 Milano, Italy
{piero.fraternali,davide.martinenghi,marco.tagliasacchi}@polimi.it

Abstract. This paper reports on recent findings regarding diversity queries over objects embedded in a low-dimensional vector space. Among the many contexts of interest, we mention spatial Web objects, which are abundant in location-based services that let users attach content to places. Typical queries aim at retrieving the best set of relevant objects that are well distributed over a region of interest. Existing methods for answering diversified top-k queries are too costly, as they evaluate diversity by accessing and scanning all relevant objects, even if only a small subset thereof is needed. Our proposal, named SPP, is an algorithm that, while finding exactly the same result as MMR (one of the most popular diversification algorithms), does not require retrieving all the relevant objects and, indeed, minimizes the number of accessed objects. Experiments confirm that SPP saves a significant amount of accesses while incurring a very low computational overhead.

1 Introduction

Geo-referenced data are becoming more and more available on the Web, especially after the advent of location-based services, whereby users can create content attached to places. Web spatial objects are also found in real estate directories, local news aggregators, image sharing sites, and travel services. Queries that require the uniform coverage of a region through spatial scattering of results are very common. An example is that of a user moving to a new city who wants an overview of real estate offers that meet some relevance criterion (e.g., price) and cover most neighborhoods.

In this paper, we address the problem of answering *top-k diversity queries* over online data sources *covering a region of interest* by presenting a synthesis of the results described in [6]. We assume that objects are represented in a vector space and can be fetched through interfaces, common for Web data sources, granting *sorted* access either by relevance (e.g., an object property or the degree of match with the query) or by distance from a given point. Our objective is to improve performance of diversified query processing by accessing only a small number of objects that guarantee to find the best result set in terms of both relevance and diversity. This is in contrast with classical diversification techniques, which access all the objects first, and then choose the best subset of diversified objects. As an example, consider a real estate query: a sample search in a commercial service for flats in London between £200,000 and £300,000 returned 60,000+ results; if the user wants to browse just a few dozens of them in diverse neighborhoods, we should access and present a number of objects proportional

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

to the user's wishes, scattered throughout the London region, without accessing all the 60,000+ relevant flats. In addition, we rely solely on the presence of sorted access methods based on relevance and distance, without requiring the knowledge of the specific index structures being used, as these typically reside on remote third-party services.

Top-k diversity queries over a vector space require a mix of techniques from top-k query processing and result diversification. As in top-k query processing [9, 11], the cost model of access methods requires minimizing the number of fetched objects. To this end, a threshold (upper bound) on the value of an objective function that quantifies both score and diversity is maintained. As more candidate objects are accessed via *probe queries*, the upper bound decreases until the guarantee is reached that no unseen object can lead to a value of the objective function better than the one determined by the already retrieved objects. Unlike in top-k query processing, the objective function of diversity queries cannot be computed on individual objects, as it uses a diversity measure (e.g., spatial scatter) that requires comparing the next object with the previously ranked ones. Existing diversification methods [2, 7] solve the issue by comparing *all* the relevant objects (i.e., the results of the user query) with the objects that have been top-ranked so far, thus materializing and scanning all of them several times [2].

We propose a novel approach, which integrates the notion of probe queries into the framework of result diversification, providing on-the-fly construction of top-k result sets that are both relevant and diverse, by using only sorted access methods and without fetching all relevant objects. Our approach works as follows. The top-k set is built incrementally, adding each time the object that maximizes an objective function based on both relevance and diversity. At each step, we probe the vector space by issuing distance-based queries at suitable points, called *probing locations*, that are likely to lie close to the best objects. We may alternatively use score-based access to retrieve objects with high relevance. Based on the retrieved objects, we maintain an upper bound on the value of the objective function that can be attained by using the unseen objects. The currently best object is added to the set when the value of the objective function determined by its inclusion is at least as high as the upper bound. Note that the choice of probing location and the alternation of score-based and distance-based access (akin to a pulling strategy [11]) exploits the geometry of the vector space. The proposed approach introduces efficiency without compromising the quality of diversification wrt. the best known general-purpose diversification algorithms.

2 Preliminaries

Consider a query q selecting a finite set $\mathcal{O}$ of objects. The *relevance* of an object $o \in \mathcal{O}$ to q is represented by a *score* $S_q(o) \in \mathbb{R}$. Let each object o also be associated with a real-valued feature vector $\mathbf{x}(o) \in \mathbb{R}^d$, denoting characteristics of the object that can be used to compute diversity. Then, *diversity* of two objects is expressed by a measure $\delta : \mathcal{O} \times \mathcal{O} \to \mathbb{R}^+$, where the value 0 indicates maximum similarity.

Let $N = |\mathcal{O}|$ denote the cardinality of the set $\mathcal{O}$, and $\mathcal{O}_K \subseteq \mathcal{O}$ a subset of K objects that are selected, e.g., to be presented to the user. We are interested in identifying a subset $\mathcal{O}_K$ that is both relevant and diverse. The diversification problem, i.e., computing the best subset $\mathcal{O}_K^*$ of $\mathcal{O}$, can be expressed as an optimization problem as follows [7]:

$$\mathcal{O}_K^* = \underset{\mathcal{O}_K \subseteq \mathcal{O}, |\mathcal{O}_K| = K}{\operatorname{argmax}} F(\mathcal{O}_K; S_q(\cdot), \delta(\cdot,\cdot)) \tag{1}$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Algorithm 1: MMR algorithm

Input: *Set of objects $\mathcal{O}$; result size K*
Output: *Selection $\mathcal{O}_K$ from $\mathcal{O}$*
Parameters: *Initialization Strategy* IS
1. $\mathcal{O}_K := \{\texttt{IS.initialObject}()\}$;
2. **while** $(|\mathcal{O}_K| < K)$
3. $\quad o^* := \underset{o \in \mathcal{O} \setminus \mathcal{O}_K}{\operatorname{argmax}} \{(1-\lambda) S_q(o) + \lambda \min_{o' \in \mathcal{O}_K} \delta(o, o')\}$;
4. $\quad \mathcal{O}_K := \mathcal{O}_K \cup \{o^*\}$;
5. **return** $\mathcal{O}_K$;

where $F(\cdot)$ is an objective function that takes into account both relevance and diversity. Solving problem (1) is NP-hard [7] for various objective functions. Hence, the need for approximate greedy algorithms, among which Maximum Marginal Relevance (MMR) [2], is one of the most popular. MMR implicitly adopts the following $F(\cdot)$:

$$F(\mathcal{O}_K) = (1-\lambda) \sum_{o \in \mathcal{O}_K} S_q(o) + \lambda \min_{o_u, o_v \in \mathcal{O}_K} \delta(o_u, o_v) \tag{2}$$

where λ is a parameter in $[0, 1]$ specifying the trade-off between relevance and diversity.

Algorithm 1 illustrates the details of MMR, which incrementally constructs $\mathcal{O}_K$ by adding, at each step, an object that is both relevant and distant from the already selected objects. The initial object is chosen according to some initialization strategy IS; typically, the object that maximizes $S_q(\cdot)$ is selected. The added object o^* (line 3 of Algorithm 1) maximizes the *diversity-weighted score* σ, defined as:

$$\sigma(o; \mathcal{O}_K) = (1-\lambda) S_q(o) + \lambda \min_{o' \in \mathcal{O}_K} \delta(o, o') \tag{3}$$

thereby maximizing also $F(\mathcal{O}_K \cup \{o\})$.

The algorithms proposed for optimizing the MMR objective function assume that all the N objects relevant to the query are retrieved and re-ranked so as to select the top K diversified elements. Therefore, all such algorithms exhibit a complexity that depends on N. For example, the overall time complexity of MMR is $O(K^2 N)$.

3 Bounded Diversification with Sorted Access Methods

The diversification problem addressed in this paper assumes that the feature vectors of the objects are contained in a finite boundary region. We consider two kinds of sorted access methods for fetching the objects:

– *Score-based access*. The set $\mathcal{O}$ is accessed sequentially in decreasing order of $S_q(\cdot)$, i.e., of relevance to the query.

– *Distance-based access*. The set $\mathcal{O}$ is accessed sequentially in increasing order of $\delta(\cdot, \mathbf{v})$, where $\mathbf{v}$ is an arbitrary vector in $\mathbb{R}^d$ called *probing location*. For example, objects are retrieved by geographical distance wrt. a given point.

Bounded diversification problems are diversification problems where the objects lie in a bounded region and can be accessed only by score or distance. We restrict our attention to the class of MMR-*correct* algorithms, defined as those deterministic algorithms

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Algorithm 2: PBMMR$(K, \mathcal{U})$

Input: *result size K; bounding region $\mathcal{U}$*
Output: *Selection $\mathcal{O}_K$ from the objects enclosed in $\mathcal{U}$*
Main vars: *set P of retrieved objects; discarded region $\mathcal{D}$; last score S_q^{last}; upper bound τ; top diversity-weighted score σ^*; top object o^**
Parameters: *Init. Strategy* IS*; Pulling Strategy* PS*; Bounding Scheme* BS

1. $\mathcal{O}_K := \{$IS.initialObject()$\}$;
2. $\mathcal{D} := \emptyset$; $S_q^{\text{last}} := 1$; $P := \mathcal{O}_K$;
3. **while** ($|\mathcal{O}_K| < K$)
4. $\quad \tau := \infty$; $\sigma^* = -\infty$;
5. $\quad$ **if** ($P \setminus \mathcal{O}_K \neq \emptyset$) **then** $o^* = \underset{o \in P \setminus \mathcal{O}_K}{\operatorname{argmax}} \sigma(o; \mathcal{O}_K)$; $\sigma^* = \sigma(o^*; \mathcal{O}_K)$;
6. $\quad$ **while** ($\sigma^* < \tau$ **and** $\mathcal{D} \subset \mathcal{U}$)
7. $\quad\quad m :=$ PS.chooseAccessMethod();
8. $\quad\quad o := m$.getNextObject();
9. $\quad\quad P := P \cup \{o\}$;
10. $\quad\quad$ **if** ($\sigma(o; \mathcal{O}_K) > \sigma^*$) **then** $\sigma^* = \sigma(o; \mathcal{O}_K)$; $o^* := o$;
11. $\quad\quad (\mathcal{D}, S_q^{\text{last}}) := m$.updateDiscardedRegionAndScore$(\mathcal{D}, S_q^{\text{last}})$;
12. $\quad\quad \tau :=$ BS.updateBound$(P, \mathcal{D}, S_q^{\text{last}})$;
13. $\quad \mathcal{O}_K := \mathcal{O}_K \cup \{o^*\}$;
14. **return** $\mathcal{O}_K$;

that select at each step exactly the same object as MMR, and thus output the same result $\mathcal{O}_K^{\text{MMR}}$. A family of algorithms solving this problem is shown in Algorithm 2, which we call *Pull/Bound* MMR (PBMMR), adapted from the *Pull/Bound Rank Join* template originally introduced for the rank join problem in [11]. The algorithm selects one object per outermost iteration. The selection is made by keeping track of an upper bound τ (computed via a *bounding scheme* BS) on the best diversity-weighted score attainable by visiting unseen objects, based on the region $\mathcal{D}$ of space already explored, the best score possible S_q^{last}, and the visited objects P. At each step of the exploration, the chooseAccessMethod function of a given *pulling strategy* PS decides the access method m to use for retrieving the next object (score-based or distance-based), in the latter case also deciding which probing location to use, i.e., from which point in the vector space to start returning objects in increasing order of distance.

Theorem 1. PBMMR *is* MMR*-correct.*

4 Space Partitioning and Probing

4.1 Probing Locations

We start the illustration of the SPP algorithm by discussing the policy for determining the probing locations, i.e., the starting points used for distance-based access. Ideally, each time a distance-based access is made, one should explore the region of space that grants the highest chances to retrieve the object with the best diversity-weighted score. To this end, at each of the K iterations of the algorithm (line 3), we fix the probing locations at the most promising points of the unexplored space. Then, we use these probing locations in the iterations of the inner loop (line 6), possibly querying the same location multiple times with an increased search radius.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

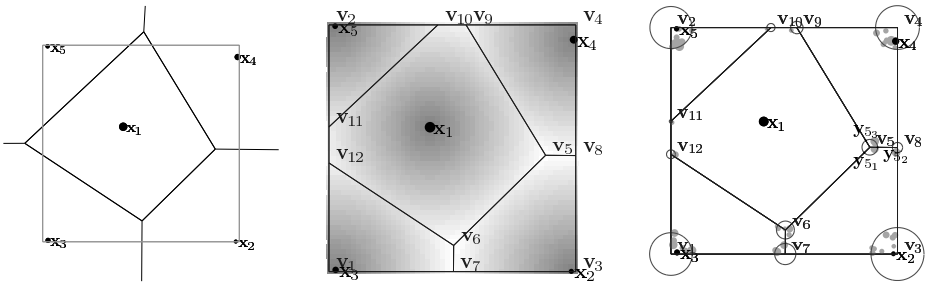


Fig. 1. Voronoi diagrams for Example 1.

At each of the K main iterations, the most promising probing locations are points that lie within the bounding region $\mathcal{U}$ and are as far as possible from *all* the objects of the current selection $\mathcal{O}_\ell$. Let $\mathcal{X} = \{\mathbf{x}(o) \in \mathbb{R}^d | o \in \mathcal{O}_\ell\}$ denote the set of points corresponding to the current selection $\mathcal{O}_\ell$. Then, the probing locations can be defined as the local maxima of the function f that expresses the distance of a point $\mathbf{x} \in \mathcal{U}$ from the closest object in the current selection: $f(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$.

An effective procedure for determining probing locations when $\mathcal{U}$ is a bounded polyhedron is provided by Theorem 2, which ensures that the local maxima of $f(\mathbf{x})$ lie in a subset of the vertices of the *bounded Voronoi diagram* $\text{Vor}(\mathcal{X},\mathcal{U})$ [3] of the points $\mathcal{X}$ corresponding to the current selection $\mathcal{O}_\ell$, obtained by restricting the conventional Voronoi diagram $\text{Vor}(\mathcal{X})$ to the region $\mathcal{U}$.

Theorem 2. *If $\mathbf{x}^* \in \mathcal{U}$ is a local maximum of $f(\mathbf{x})$ then, $\mathbf{x}^*$ is a vertex of $\text{Vor}(\mathcal{X},\mathcal{U})$.*

Theorem 2 allows us to efficiently find a superset of the local maxima by constructing $\text{Vor}(\mathcal{X},\mathcal{U})$ and enumerating its vertices $\mathbf{v}_u$, $u = 1, \ldots, V$. Vertices that are not local maxima can be disregarded.

Example 1. The left graph of Figure 1 illustrates an example when $d = 2$ and $\ell = 5$ objects have already been determined within a bounding rectangle $\mathcal{U}$. The corresponding points $\mathbf{x}_1, \ldots, \mathbf{x}_5$ define the Voronoi diagram $\text{Vor}(\mathcal{X})$. Note that $\mathcal{C}_1$ of $\mathbf{x}_1$ is bounded, whereas all the other cells are unbounded. The corresponding bounded Voronoi diagram $\text{Vor}(\mathcal{X},\mathcal{U})$ is represented in the middle graph of Figure 1. Vertices $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ and $\mathbf{v}_4$ correspond to the original vertices of $\mathcal{U}$. Out of the four vertices of $\text{Vor}(\mathcal{X})$, only two are retained (i.e., $\mathbf{v}_5$ and $\mathbf{v}_6$), as the other ones are outside $\mathcal{U}$. The remaining vertices ($\mathbf{v}_7$ to $\mathbf{v}_{12}$) are due to intersections between $\text{Vor}(\mathcal{X})$ and the edges of $\mathcal{U}$. The shading indicates the distance from the closest point in $\mathcal{X}$, where brighter indicates larger distance. Such a distance is maximized at the vertices, as confirmed by Theorem 2.

4.2 Bounding scheme

We now turn to the computation of the upper bound τ in a given running state. To exemplify a running state, the right graph of Figure 1 shows the discarded region $\mathcal{D}$ as a set of hyperspheres (in red) enclosing the previously accessed objects (shown as light red

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

discs with sizes proportional to the scores). Note that Vor($\mathcal{X}$,$\mathcal{U}$) and the corresponding probing locations are updated each time a new selected object is added by PBMMR.

The unseen objects retrievable with the next distance-based access belong to the set $\mathcal{Z} = \mathcal{U} \setminus \mathcal{D}$, which leaves out each explored hypersphere Σ_u centered in $\mathbf{v}_u$, $u = 1, \ldots, V$. Indeed, after an object at a distance r_u is extracted from $\mathbf{v}_u$, no new object can lie closer than that to $\mathbf{v}_u$. A tight upper bound can be found as follows

$$\tau = (1 - \lambda) S_q^{\text{last}} + \lambda \max_{\mathbf{x} \in \mathcal{Z}} \min_{\mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\| \tag{4}$$

Theorem 3 provides an effective computation procedure for (4).

Theorem 3. *The point $\mathbf{x}^* \in \mathcal{Z}$ that maximizes the minimum distance from all the points in $\mathcal{X}$ is a vertex of the convex hull of $\mathcal{P}_i \setminus \mathcal{D}$, where $\mathcal{P}_i$ is one of the cells of Vor($\mathcal{X}$,$\mathcal{U}$).*

Thanks to Theorem 3, τ as of (4) can be computed by enumerating the cells of Vor($\mathcal{X}$,$\mathcal{U}$) and, for each cell diminished by $\mathcal{D}$, the vertices of its convex hull. Equivalently, in 2D, we can enumerate each vertex $\mathbf{v}_u$ of Vor($\mathcal{X}$,$\mathcal{U}$), and find the intersections of the circumference of Σ_u with the edges or other circumferences.

Example 2. With reference to Figure 1, let us consider $\mathbf{v}_5$. We have $p_5 = 3$ vertices (namely, $\mathbf{v}_6$, $\mathbf{v}_8$, and $\mathbf{v}_9$) connected to $\mathbf{v}_5$ through edges. The circumference Σ_5 centered in $\mathbf{v}_5$ intersects such edges in three points: $\mathbf{y}_{5_1}$, $\mathbf{y}_{5_2}$, and $\mathbf{y}_{5_3}$.

The appropriateness of the bounding scheme stems from tightness of the upper bound, in the sense that the value of the bound can be achieved in some hypothetical continuation of the instance being explored, i.e., for some assignment of admissible location and score to the unseen objects.

Theorem 4. *The bounding scheme* (4) *is tight.*

4.3 Pulling strategy

The pulling strategy determines how to fetch the next object, alternating between distance-based and score-based access (when available). The pulling strategy can be as simple as a *round-robin* (RR) scheduling, whereby distance- and score-based access are alternated, and probing locations are uniformly explored. Tightness of the bounding scheme and a RR strategy are sufficient to guarantee a form of instance optimality. Let $\mathcal{A}^{\text{Vor}}$ be the class of MMR-correct, deterministic bounded diversification algorithms that can discover objects by both score-based and distance-based access using the vertices of Vor($\mathcal{X}$,$\mathcal{U}$) as probing locations.

Theorem 5. *Algorithm 2 with tight bounding scheme* (4) *and a RR pulling strategy is instance-optimal wrt. $\mathcal{A}^{Vor}$.*

In order to further decrease $sumDepths$ (i.e., the overall amount of accessed objects), a *potential adaptive* (PA) pulling strategy can be devised so as to lower the upper bound τ more quickly. PA works as follows: when a distance-based access is to be made, the probing location $\mathbf{v}_{u^*}$ is selected with $u^* = \operatorname{argmax}_{u=1,\ldots,V} \tau_u$. Conventionally, ties are broken in favor of the probing location with the least depth, then the one with the least index in $1, \ldots, V$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Theorem 6. *Let A^{RR} and A^{PA} be algorithms in $\mathcal{A}^{Vor}$ using tight bounding scheme* (4) *with the RR and the PA pulling strategies, respectively. Then $sumDepths(A^{PA}, I) \leq sumDepths(A^{RR}, I)$ for all bounded diversification problems I.*

When both distance-based and score-based access are available, we seek the one that reduces τ at a faster rate. To this end, we compute the partial derivatives of τ wrt. the number of fetched objects. Let n^S denote the number of objects retrieved by means of score-based access, and n_u, $u = 1, \ldots, V$, the number of objects retrieved by distance-based access from probing location $\mathbf{v}_u$. Score-based access is preferred if $\left|\frac{\partial \tau}{\partial n^S}\right| > \left|\frac{\partial \tau}{\partial n_{u^*}}\right|$. These partial derivatives can be either computed exactly if the distribution is known, or approximated by using, e.g., linear predictors.

We define SPP as the instance of Algorithm 2 that uses the tight bounding scheme (4) and the PA pulling strategy.

5 Related work

Result Set Diversification. A general formulation of diversification is introduced in [7]. Existing approaches (surveyed in [5]) rerank relevant results to introduce diversity. Unlike SPP, these approaches scan all the n candidate results. Diversification in multiple dimensions is addressed in [4], where the problem is reduced to MMR by collapsing diversity dimensions in one composite similarity function. The recent work [1] examines diversity-aware search under the angle of performance. Unlike [1], our work addresses diversification in a different scenario, where objects are embedded in a vector space, and exploits the geometry in order to limit the number of accessed objects.

We have focused on algorithms that can extract, on the fly, the top k relevant and diversified objects from data sources providing sorted access methods (by score or distance). This class of algorithms is very general and applies to all those cases where extracting *all* the relevant objects and then scanning them for diversification is impractical (e.g., mobile queries). Many general-purpose diversification algorithms exist [15]. For some of these (e.g., Motley), the same formal apparatus as PBMMR is clearly applicable. However, other algorithms described in [15] (e.g., GMC) fall outside the PBMMR paradigm, in that they require knowing all the objects beforehand.

Spatial Diversification. Spatial diversification was originally introduced by [14]. The scattered ranking approach exploits the geometry of the metric space to reduce the number of operations for creating the ranking; however, the proposed algorithms access all the N relevant points. The experiment with Mechanical Turk in [13] shows that users prefer spatially diversified rankings over undiversified ones.

Top-k Query Processing. The main design dimensions and tools for top-k queries are surveyed in [8]. In [10], rank join is extended to objects in a d-dimensional space, with the aim of finding the best combinations of objects with high score that are close to a given point and to each other. The technique in [10] also uses geometry-driven bounds, but for a different problem (rank join) and geometry than ours.

6 Conclusions and Future Work

We have addressed the problem of efficiently diversifying the results of top-k queries over spatial objects contained in a bounded region when only sorted access methods

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

based on distance and/or score are permitted. Our work on top-k has included the following contributions: *i) Bounded diversification with sorted access methods* is introduced for the first time and defined formally. *ii)* The *Pull/Bound Maximum Marginal Relevance* (PBMMR) family of algorithms is illustrated, which exploits spatial probing locations and the adaptive alternation of score-based and distance-based access to reduce the number of fetched objects. *iii)* An instance of PBMMR, called *Space Partitioning and Probing* (SPP), is presented, whose pulling strategy uses a tight upper bound. *iv)* SPP is shown to attain the same diversification quality and exactly the same output as MMR, the most popular result diversification algorithm, but accessing only a fraction of the objects. *v)* Experiments, omitted here for brevity, show that, with a negligible computational overhead, SPP accesses in typical conditions less than 20% of the objects (less than 2% in best conditions), a substantial gain over past work on spatial scatter queries [14], which accesses all objects. Future work includes extending SPP to arbitrary joins of data sources. Also, we plan to tackle the possible presence of uncertainty in the data as was done in [12] for top-k queries.

Acknowledgements The authors acknowledge support from *i)* the EC's FP7 "CUbRIK" project, *ii)* the ERC "Search Computing" project, *iii)* the Italian (MIUR) "EASE" project.

References

1. A. Angel and N. Koudas. Efficient diversity-aware search. In *SIGMOD Conference*, pages 781–792, 2011.
2. J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, pages 335–336, 1998.
3. M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.
4. Z. Dou et al. Multi-dimensional search result diversification. In *WSDM '11*, pages 475–484, 2011.
5. M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Rec.*, 39(1):41–47, 2010.
6. P. Fraternali, D. Martinenghi, and M. Tagliasacchi. Top-k bounded diversification. In *SIGMOD Conference*, 2012.
7. S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW '09*, pages 381–390, 2009.
8. I. F. Ilyas, G. Beskales, and M. A. Soliman. A survey of top-k query processing techniques in relational database systems. *ACM Comput. Surv.*, 40(4), 2008.
9. A. Marian, N. Bruno, and L. Gravano. Evaluating top- queries over web-accessible databases. *ACM Trans. Database Syst.*, 29(2):319–362, 2004.
10. D. Martinenghi and M. Tagliasacchi. Proximity rank join. *PVLDB*, 3(1):352–363, 2010.
11. K. Schnaitter and N. Polyzotis. Evaluating rank joins with optimal cost. In *PODS*, pages 43–52, 2008.
12. M. A. Soliman, I. F. Ilyas, D. Martinenghi, and M. Tagliasacchi. Ranking with uncertain scoring functions: semantics and sensitivity measures. In *SIGMOD Conference*, pages 805–816, 2011.
13. J. Tang and M. Sanderson. Evaluation and user preference study on spatial diversity. In *ECIR*, pages 179–190, 2010.
14. M. J. van Kreveld et al. Multi-dimensional scattered ranking methods for geographic information retrieval. *GeoInformatica*, 9(1):61–84, 2005.
15. M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. T. Jr., and V. J. Tsotras. On query result diversification. In *ICDE*, pages 1163–1174, 2011.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

A Query Reformulation Framework for P2P OLAP*

Matteo Golfarelli[1], Federica Mandreoli[2],
Wilma Penzo[1], Stefano Rizzi[1], and Elisa Turricchia[1]

[1] DEIS - Univ. of Bologna, V.le Risorgimento 2, Bologna, Italy
[2] DII - Univ. of Modena and Reggio Emilia, Via Vignolese 905/b, Modena, Italy

Abstract. The idea of collaborative business intelligence is to extend the decision-making process beyond the company boundaries thanks to cooperation and data sharing with other companies and organizations. In this direction, we propose a query reformulation framework based on a P2P network of heterogeneous peers, each exposing OLAP query answering functionalities aimed at sharing business information. In our framework, an OLAP query expressed on a peer is reformulated on other peers by relying on a set of mappings between the multidimensional schemata of peers. In this extended abstract we sketch the user interaction scenario we envision and briefly discuss each phase of the reformulation process.

Keywords: business intelligence, OLAP, P2P architectures, query reformulation

1 Introduction

In the current changeable and unpredictable market scenarios, the needs of decision makers are rapidly evolving as well. This gave rise to a new generation of business intelligence (BI) systems, often labeled as *BI 2.0*. One of the key features of BI 2.0 is the ability to become collaborative and extend the decision-making process beyond the boundaries of a single company [12]. Users need to transparently access information anywhere it can be found, by locating it through a semantic process and performing integration on the fly. This is particularly relevant in inter-business collaborative contexts where companies organize and coordinate themselves to share opportunities, respecting their own autonomy and heterogeneity but pursuing a common goal. For instance, this is the case of companies in a supply chain, or local health-care departments that collaborate to enable effective analysis of epidemics and health-care costs [6]. In such a complex and distributed business scenario, traditional BI systems —that were born to support stand-alone decision-making— are no longer sufficient to maximize the effectiveness of monitoring and decision making processes.

To fill this gap, we envision a peer-to-peer data warehousing architecture called *Business Intelligence Network* (BIN). A BIN is an architecture for sharing

* An extended version of this work is published in [6].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

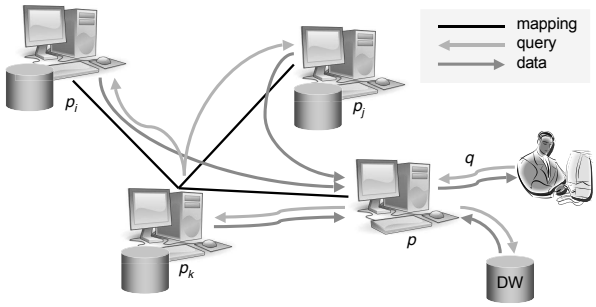


Fig. 1. Interaction scenario for a BIN

BI functionalities across a dynamic and collaborative network of heterogeneous and autonomous peers. Each peer is equipped with an independent data warehouse system, that relies on a local multidimensional schema to represent the peer's view of the business and exposes OLAP query answering functionalities aimed at sharing business information, in order to enhance the decision making process and create new knowledge. The main benefit of the BIN approach stands in the ability to efficiently manage inter-company processes and safely sharing management information besides operational information [5].

The core idea of a BIN is that of enabling users to transparently access business information distributed over the network. A typical interaction sequence is the following (Figure 1):

1. A user formulates an OLAP query q by accessing the local multidimensional schema exposed by her peer, p.
2. Query q is processed locally on the data warehouse of p.
3. At the same time q is forwarded to the network.
4. Each involved peer locally processes the query on its data warehouse and returns its results to p.
5. The results are integrated and returned to the user.

The local multidimensional schemata of peers are typically heterogeneous. So, during distributed query processing, before a query issued on a peer can be forwarded to the network it must be first *reformulated* according to the multidimensional schemata of the source peers. Data are then extracted from each *source* peer and are mapped onto the schema of the querying (*target*) peer.

In line with the approach adopted in *Peer Data Management Systems* (PDMSs) [8], query reformulation in a BIN is based on *semantic mappings* that mediate between the different multidimensional schemata exposed by two peers, i.e., they describe how the concepts in the multidimensional schema of the source peer map onto those of the target peer. Peers establish semantic mappings by exchanging their local schemata and applying a schema-matching algorithm [11]. Direct mappings cannot be realistically defined for all the possible couples of peers. So, to enhance information sharing, a query q issued on p is forwarded to the network by first sending it to the neighborhood of p; then, each peer in

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

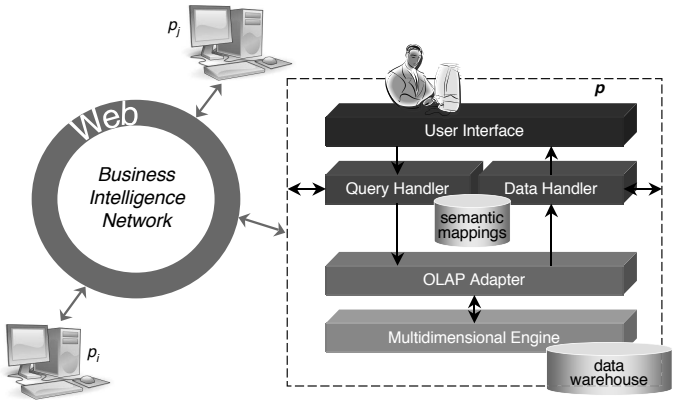


Fig. 2. Envisioned architecture for a BIN

this neighborhood in turn sends q to its neighborhood, and so on.[3] In this way, q undergoes a chain of reformulations along the peers it reaches, and results are collected from any peer that is connected to p through a path of semantic mappings.

The approach outlined above is reflected by the internal architecture of each peer, sketched in Figure 2, whose components are:

1. *User Interface*. A web-based component that manages bidirectional interaction with users, who use it to visually formulate OLAP queries on the local multidimensional schema and explore query results.
2. *Query Handler*. This component receives an OLAP query from either the user interface or a neighboring peer on the network, sends that query to the OLAP adapter to have it locally answered, reformulates it onto the neighborhood (using the available semantic mappings), and transmits it to the peers in that neighborhood.
3. *Data Handler*. When processing a locally-formulated query, the data handler collects query results from the OLAP adapter and from the other peers, integrates them, and returns them to the user interface. When processing a query formulated on some other peer p, the data handler just collects local query results from the OLAP adapter and returns them to p.
4. *OLAP Adapter*. This component adapts queries received from the query handler to the querying interface exposed by the local multidimensional engine.
5. *Multidimensional Engine*. It manages the local data warehouse according to the multidimensional schema representing the peer's view of the business, and provides MDX-like query answering functionalities.

Query answering in a BIN architecture poses several research challenges, ranging from languages and models for semantic mediation to query reformulation issues and proper techniques and data structures for the query processing

[3] To improve query processing efficiency, query routing strategies to select a subset of neighboring peers for query reformulation could be employed [11].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

phase. In particular, much work on query reformulation has been done in the context of PDMSs [8] and relational databases [4], however those results are not directly applicable in the OLAP scenario presented by the BIN, that poses additional challenges due to the presence of aggregation and to the possibility of having information represented at different granularities in each peer. The framework for query reformulation in a BIN we outline in this extended abstract relies on the translation of semantic mappings and queries towards the underlying relational schemata. Mappings between the schemata of peers are expressed using predicates that are specifically tailored for the multidimensional model; to overcome possible differences in data formats and information granularities, mappings can be associated with transcoding functions. The query reformulation algorithm is correct, with polynomial complexity [6], and can be safely used to implement chains of reformulations as required in the BIN setting.

In the following sections the main aspects of the query reformulation process will be intuitively discussed based on a working example.

2 Mapping Language

Reformulation of OLAP queries first of all requires a language for properly expressing the semantic mappings between each couple of neighboring peers. The language used in a BIN accommodates the peculiar characteristics of the multidimensional model, on which the representation of business information at each peer is founded. It expresses how the multidimensional schema $\mathcal{M}_s$ of a source peer s maps onto the multidimensional schema $\mathcal{M}_t$ of a target peer t using the *mapping predicates* explained below. In general, a mapping establishes a semantic relationship from one or more concepts (either measures or attributes) of $\mathcal{M}_s$ to one or more concepts of $\mathcal{M}_t$, and enables a BIN query formulated on $\mathcal{M}_t$ to be reformulated on $\mathcal{M}_s$. Optionally, a mapping involving attributes can be annotated with a *transcoding function* that specifies how values of the target concepts can be obtained from values of the source concepts. A transcoding function can be a standard database function (e.g., `substring`) shared by all peers, as well as a function owned by a peer and made available to its neighbors by attaching it to query messages. If this function is available, it is used to increase the reformulation effectiveness, e.g., by enabling data returned by the source and target peers to be integrated.

- The `same` predicate states that whenever a given measure is asked in a query on $\mathcal{M}_t$ using a given aggregation operator, it can be rewritten as a given expression involving the measures in $\mathcal{M}_s$.
- The `equi-level` predicate states that two sets of attributes of $\mathcal{M}_t$ and $\mathcal{M}_s$, respectively, have the same semantics and granularity. Optionally, it can be annotated with a transcoding function that establishes a one-to-one relation between tuples of values of the two sets of attributes.
- The `roll-up` predicate states that a set of attributes of $\mathcal{M}_t$ aggregates a set of attributes of $\mathcal{M}_s$. Optionally, it can be annotated with a transcoding

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

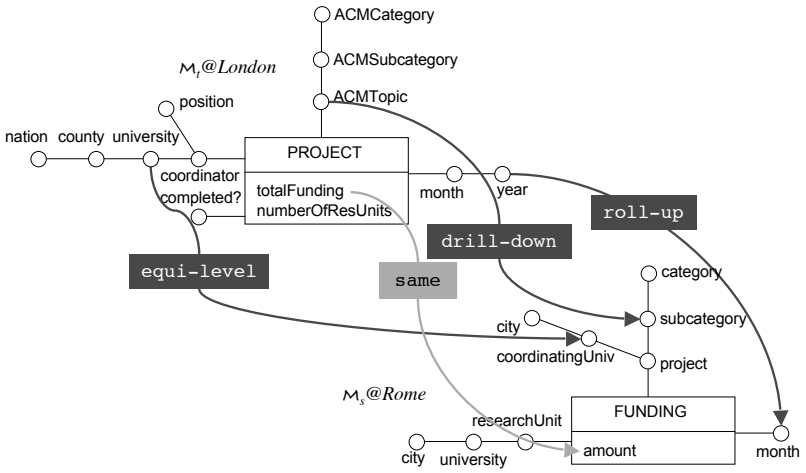


Fig. 3. Multidimensional schemata of related facts at two peers

function that establishes a many-to-one relation between tuples of values of the two sets of attributes.

– The `drill-down` predicate states that a set of attributes of $\mathcal{M}_t$ disaggregates a set of attributes of $\mathcal{M}_s$. Optionally, it can be annotated with a transcoding function that establishes a one-to-many relation between tuples of values of the two sets of attributes.

Example 1. Consider a BIN for sharing information about funded research projects among European nations. Figure 3 shows the multidimensional schemata of related facts at the peers in London and Rome, using the Dimensional Fact Model notation [7]; small circles represent attributes, while measures are listed inside the fact boxes. Figure 3 also shows some of the mappings that can be defined to reformulate queries expressed in London (target peer) according to the schema adopted in Rome (source peer). As examples of transcodings, consider the function that associates each topic in the ACM classification with a subcategory and the one that associates each month with its year, used to annotate mappings ACMTopic `drill-down` subcategory and year `roll-up` month, respectively. Similarly, the `same` mapping between totalFunding and amount is annotated with an expression that converts euros into pounds using the exchange rate 0.872. □

3 Inter-Peer Query Reformulation

Reformulation takes as input an OLAP query on a target schema $\mathcal{M}_t$ as well as the mappings between $\mathcal{M}_t$ and the schema of one of its neighbor peers, the source schema $\mathcal{M}_s$, to output an OLAP query that refers only to $\mathcal{M}_s$. The reformulation framework we adopt is based on a relational setting where the multidimensional schemata, OLAP queries, and semantic mappings at the OLAP level are translated to the relational model. As to multidimensional schemata,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

@London: ProjectFT(coordinator,completed,ACMTopic,month,totalFunding,numberOfResUnits);
CoordinatorDT(coordinator,university,county,nation,position);
ACMTopicDT(ACMTopic,ACMSubcategory,ACMCategory); MonthDT(month,year)
@Rome: FundingFT(researchUnit,project,month,amount); ResearchUnitDT(researchUnit,university,city);
ProjectDT(project,subcategory,category,coordinatingUniv,city)

Fig. 4. Star schemata for the London and Rome peers

without loss of generality we assume that they are stored at the relational level as star schemata. As to queries, a classic logic-based syntax is adopted to express them at the relational level. As to mappings, their representation at the relational level uses a logical formalism typically adopted for schema mapping languages, i.e., *source-to-target tuple generating dependencies* (s-t tgd's) [3]. A query is then reformulated starting from its relational form on a star schema, using the mappings expressed as s-t tgd's. A detailed explanation of the reformulation process can be found in [6], together with the reformulation algorithm and its proof of correctness.

Example 2. Consider the OLAP query q asking, at the London peer, for the total funding of projects about each subcategory of category 'Information Systems' in 2011. Reformulating this query onto the Rome peer requires:

1. Translating the multidimensional schemata at both London and Rome into star schemata, which produces the result shown in Figure 4.
2. Translating q into a relational query on the London star schema:

$$q : \pi_{\mathsf{ACMSubcategory},\mathtt{SUM}(\mathsf{totalFunding})}\sigma_{(\mathsf{year}=\text{'2011'},\mathsf{ACMCategory}=\text{'Inf. Sys.'})}\chi_{London}$$

   where χ_{London} denotes the star join made over the London star schema.
3. Translating the mappings involved into s-t tgd's. For this query, the involved mappings (each annotated with its transcoding or expression) are:

$$\begin{aligned} \mathsf{ACMCategory}\ \mathtt{equi\text{-}level}_{id}\ \mathsf{category} \\ \mathsf{ACMSubcategory}\ \mathtt{equi\text{-}level}_{id}\ \mathsf{subcategory} \\ \mathsf{year}\ \mathtt{roll\text{-}up}_{\mathtt{YearOf}}\ \mathsf{month} \\ \langle \mathsf{totalFunding}, \mathtt{SUM} \rangle\ \mathtt{same}_{\mathsf{amount}*0.872}\ \mathsf{amount} \end{aligned}$$

   where id is the identity function and the `YearOf` transcoding associates each month in the Rome format to its year in the London format.

Using the reformulation algorithm proposed in [6], q is then translated into the following query over the Rome schema:

$$q' : \pi_{\mathsf{subcategory},\mathtt{SUM}(\mathsf{amount}*0.872)}\sigma_{(\mathtt{YearOf}(\mathsf{month})=\text{'2011'},\mathsf{category}=\text{'Inf. Sys.'})}\chi_{Rome}$$

Remarkably, in this case reformulation is *compatible*, i.e., it fully preserves the semantics of q. When a compatible reformulation is used, the results returned by the source peer do *exactly* match with q so they can be seamlessly integrated with those returned by the target peer. □

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

4 Intra-Peer Reformulation

In general, a BIN query (either directly formulated by a user or reformulated across the network) cannot be directly executed on the peer local multidimensional engine, because of the language and expressiveness gap between the query handler and the local multidimensional engine. The OLAP adapter is in charge of bridging this gap by supporting intra-peer reformulation of BIN queries, so as to complete the reformulation process. Assuming that the de-facto standard MDX is the querying language of the local multidimensional engine, intra-peer reformulation must deal with the presence of transcodings in the query group-by set, and must properly manage non-distributive aggregation operators. From the reformulation point of view, this amounts to solving a problem of *query rewriting using views* [9], where the set of views is made of all the possible queries that the engine supports. In particular, given the relational translation q of a BIN query, we have to find a local query q^{loc} that refers to one MDX result set and is equivalent to q.

Example 3. The query q' shown in Example 2 cannot be directly formulated in MDX because it involves the `YearOf` transcoding. The SQL for the corresponding local query is

```
SELECT      RS.subcategory, SUM(RS.amount*0.872)
FROM        ResultSet RS, YearOf YO
WHERE       RS.month = YO.month AND YO.year = '2011'
GROUP BY    RS.subcategory;
```

where YearOf is a lookup table for the `YearOf` transcoding and ResultSet stores the result of the following MDX query:

```
SELECT {[Measures].[amount]} ON COLUMN,
       {NonEmptyCrossJoin([Month].[month].Members,
                          [Project].[Inf.Sys.].Children)} ON ROWS
FROM [Funding]
```

□

5 Conclusions and Related Works

Supporting the sharing of information for decision-making processes is a challenging task that lays the foundations for BI 2.0. The BIN architecture and the query reformulation framework we proposed is a first, significant step in this direction. Noticeably, despite the relevance of the problem, only a few works in the literature are specifically focused on strategies for data warehouse integration and federation. Indeed, in this context, problems related to data heterogeneity are usually solved by ETL processes that read data from several data sources and load them in a single repository. While this centralized architecture may fit the needs of stand-alone companies, it is hardly feasible in the context of a BIN, where the dynamic nature of the business network, together with the independence and autonomy of peers, call for more sophisticated solutions. See [1] for

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

a discussion of the benefits of a peer-to-peer architecture for data warehousing and [13] for a description of the issues arising in collaborative BI systems.

In the context of a federated data warehouse architecture, [14] describes two methods for integrating dimensions belonging to different data marts, but the problem of how to define mappings between concepts is not considered. The work proposed in [2] presents a complete algorithm for matching multidimensional structures, but the data-related aspects are not considered, and no model is provided to formalize the mapping predicates. Another work centered on interoperability issues is [10]; since it proposes specific techniques to deal with measures only, it cannot be used to completely solve a typical aggregate query. The work that is most closely related to ours is [15]; though some goals are shared with our approach, there are important differences: peers' autonomy is not preserved and the problem of chains of reformulation is not faced.

References

1. Abiteboul, S.: Managing an XML warehouse in a P2P context. In: Proc. CAiSE. pp. 4–13. Klagenfurt, Austria (2003)
2. Banek, M., Vrdoljak, B., Tjoa, A.M., Skocir, Z.: Automated integration of heterogeneous data warehouse schemas. IJDWM 4(4), 1–21 (2008)
3. ten Cate, B., Kolaitis, P.G.: Structural characterizations of schema-mapping languages. Commun. ACM 53(1), 101–110 (2010)
4. Cohen, S., Nutt, W., Sagiv, Y.: Rewriting queries with arbitrary aggregation functions using views. TODS 31(2), 672–715 (2006)
5. Golfarelli, M., Mandreoli, F., Penzo, W., Rizzi, S., Turricchia, E.: BIN: Business intelligence networks. In: Business Intelligence Applications and the Web: Models, Systems and Technologies, pp. 244–265. IGI Global (2011)
6. Golfarelli, M., Mandreoli, F., Penzo, W., Rizzi, S., Turricchia, E.: OLAP query reformulation in peer-to-peer data warehousing. Inf. Syst. 37(5), 393–411 (2012)
7. Golfarelli, M., Rizzi, S.: Data Warehouse design: Modern principles and methodologies. McGraw-Hill (2009)
8. Halevy, A.Y., Ives, Z.G., Madhavan, J., Mork, P., Suciu, D., Tatarinov, I.: The Piazza peer data management system. IEEE TKDE 16(7), 787–798 (2004)
9. Halevy, A.: Answering queries using views: A survey. VLDBJ 10(4), 270–294 (2001)
10. Kehlenbeck, M., Breitner, M.H.: Ontology-based exchange and immediate application of business calculation definitions for online analytical processing. In: Proc. DaWaK. pp. 298–311. Linz, Austria (2009)
11. Mandreoli, F., Martoglia, R., Penzo, W., Sassatelli, S.: Data-sharing P2P networks with semantic approximation capabilities. IEEE Internet Computing 13(5), 60–70 (2009)
12. Raden, N.: Business intelligence 2.0: Simpler, more accessible, inevitable. http://intelligent-enterprise. informationweek.com (2007)
13. Rizzi, S.: Collaborative business intelligence. In: Business Intelligence - First European Summer School, pp. 186–205. Springer (2012)
14. Torlone, R.: Two approaches to the integration of heterogeneous data warehouses. Distributed and Parallel Databases 23(1), 69–97 (2008)
15. Vaisman, A., Espil, M.M., Paradela, M.: P2P OLAP: Data model, implementation and case study. Inf. Syst. 34(2), 231–257 (2009)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Efficient Query Answering over *Datalog* with Existential Quantifiers*

Nicola Leone, Marco Manna, Giorgio Terracina, and Pierfrancesco Veltri

Department of Mathematics, University of Calabria, Italy
{leone,manna,terracina,veltri}@mat.unical.it

Abstract. This paper faces the problem of answering conjunctive queries over *Datalog* programs allowing existential quantifiers in rule heads. Such an extension of *Datalog* is highly expressive, enables easy yet powerful ontology-modelling, but leads to undecidable query answering in general. To overcome undecidability, we first define *Shy*, a subclass of *Datalog* with existential quantifiers preserving not only decidability but also the same complexity of query answering over *Datalog*. Next, we design and implement a bottom-up evaluation strategy for *Shy* programs. Our computation strategy includes a number of optimizations resulting in DLV$^{\exists}$, a powerful reasoner over *Shy* programs. Finally, we carry out an experimental analysis comparing DLV$^{\exists}$ with some state-of-the-art systems for ontology-based query answering. The results confirm the effectiveness of DLV$^{\exists}$, which outperforms all other systems in the benchmark.

1 Introduction

In the field of data and knowledge management, *query answering over ontologies* (QA) is becoming more and more a challenging task [9,6,18]. In this context, a conjunctive query (CQ) q is not merely evaluated on a extensional relational database D, but over a logical theory combining D with an ontology Σ describing rules for inferring intensional knowledge from D. A key issue here is the design of the language provided for specifying Σ. It should balance expressiveness and complexity, and in particular it should possibly be: (1) intuitive and easy-to-understand; (2) QA-decidable; (3) efficiently computable; (4) expressive enough; and (5) suitable for an efficient implementation. In this regard, Datalog$^{\pm}$ [6], the family of *Datalog*-based languages recently proposed for tractable QA, is arousing increasing interest. This family, generalizing well known ontology specification languages, is mainly based on *Datalog*$^{\exists}$, the natural extension of *Datalog* [1] that allows $\exists$-quantified variables in rule heads. For example, the rules $\exists$`Y father(X,Y)` $\leftarrow$ `person(X)` and `person(Y)` $\leftarrow$ `father(X,Y)` state that if `X` is a person, then `X` must have a father `Y`, which has to be a person as well. However, even if all known QA-decidable Datalog$^{\pm}$ languages enjoy the simplicity of *Datalog* and are endowed with properties that are desired for ontology languages, none of them fully satisfy conditions (1)–(5) above (see Section 5).

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

In this work, we single out *Shy*, a new powerful, yet QA-decidable *Datalog*$^{\exists}$ class that combines positive aspects of different Datalog$^{\pm}$ languages. Concerning properties (1)–(5) above, *Shy*: (1) inherits the simplicity and naturalness of *Datalog*; (2) is QA-decidable; (3) is efficiently computable; (4) offers good expressiveness strictly generalizing *Datalog*; and (5) is suitable for an efficient implementation. From a technical viewpoint, our contribution is as follows:

- We define *Shy*, a subclass of *Datalog*$^{\exists}$ that preserves not only decidability but also the same data (resp., combined) complexity of *Datalog* for unrestricted conjunctive queries (resp., atomic queries).
- We show that *Shy* strictly encompasses both *Datalog* and *Linear-Datalog*$^{\exists}$ [6], and that it is uncomparable to all other known *Datalog*$^{\exists}$ classes.
- We design and implement a bottom-up evaluation strategy for *Shy* programs inside the DLV system. Our computation strategy includes a number of optimizations resulting in DLV$^{\exists}$, a powerful reasoner over *Shy* programs. To the best of our knowledge, DLV$^{\exists}$ is the first system supporting the standard first-order semantics for unrestricted CQs with $\exists$-variables over ontologies using advanced properties (some of these beyond $\mathbf{AC}_0$), such as, role transitivity, role hierarchy, role inverse, and concept products [14].
- We perform an experimental analysis comparing DLV$^{\exists}$ with some state-of-the-art systems for QA. The results demonstrate that DLV$^{\exists}$ is the most effective system for QA in dynamic environments where the ontology frequently changes making pre-computations and static optimizations inapplicable.

2 The Framework

This section, after introducing *Datalog*$^{\exists}$ programs and CQs, equips such structures with a formal semantics and defines the query answering problem.

Preliminaries. Throughout this paper we denote by Δ_C, Δ_N and Δ_V, countably infinite domains of *terms* called *constants*, *nulls* and *variables*, respectively; by Δ, the union of these three domains; by t, a generic *term* in Δ; by $\mathtt{X}$ and $\mathtt{Y}$, variables; by $\mathbf{X}$ and $\mathbf{Y}$, sets of variables; by Π an alphabet of *predicate symbols* each of which, say $\mathtt{p}$, has a fixed nonnegative arity; by $\mathbf{a}$ and $\mathbf{b}$, *atoms* being expressions of the form $\mathtt{p}(t_1, \ldots, t_k)$, where $\mathtt{p}$ is a predicate symbol, and $t_1, \ldots, t_k$ is a *tuple* of terms. For a formal structure ς containing atoms, $\mathsf{atoms}(\varsigma)$ denotes the set of atoms in ς, and $\mathsf{dom}(\varsigma)$ denotes the set of terms from $\Delta_C \cup \Delta_N$ occurring in $\mathsf{atoms}(\varsigma)$. If $\mathbf{X}$ is the set of all variables of ς, then ς is also denoted by $\varsigma_{[\mathbf{X}]}$. A structure $\varsigma_{[\emptyset]}$ is called *ground*. If $T \subseteq \Delta$ and $T \neq \emptyset$, then $\mathsf{base}(T)$ denotes the set of all atoms that can be formed with predicate symbols in Π and terms from T.

A *substitution* is a mapping $\sigma : \Delta_V \to \Delta_C \cup \Delta_N$. For a set $\mathbf{X} \subseteq \Delta_V$, the application of σ to $\mathbf{X}$ is the set $\sigma(\mathbf{X}) = \{\sigma(\mathtt{X}) \mid \mathtt{X} \in \mathbf{X}\}$. Moreover, the restriction of σ to $\mathbf{X}$, is the substitution $\sigma|_{\mathbf{X}}$ from $\mathbf{X}$ to $\Delta_C \cup \Delta_N$ s.t. $\sigma'(\mathtt{X}) = \sigma(\mathtt{X})$ for each $\mathtt{X} \in \mathbf{X}$. For an atom $\mathbf{a} = \mathtt{p}(t_1, \ldots, t_k)$, $\sigma(\mathbf{a})$ denotes the atom obtained from $\mathbf{a}$ by replacing each variable $\mathtt{X}$ of $\mathbf{a}$ with $\sigma(\mathtt{X})$. For a structure ς containing atoms, we denote by $\sigma(\varsigma)$ the structure obtained by replacing each atom $\mathbf{a}$ of ς with $\sigma(\mathbf{a})$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Programs and Queries. A *rule* r is a finite expression of the form:

$$\forall \mathbf{X} \exists \mathbf{Y} \;\; \mathbf{atom}_{[\mathbf{X}' \cup \mathbf{Y}]} \leftarrow \mathbf{conj}_{[\mathbf{X}]} \tag{1}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are disjoint sets of variables (next called $\forall$-variables and $\exists$-variables, respectively), and $\mathbf{X}' \subseteq \mathbf{X}$. In the following, $\mathsf{head}(r) = \mathbf{atom}_{[\mathbf{X}' \cup \mathbf{Y}]}$ and $\mathsf{body}(r) = \mathsf{atoms}(\mathbf{conj}_{[\mathbf{X}]})$. If $\mathsf{body}(r) = \emptyset$, then r is also called *fact*. A *Datalog*$^{\exists}$ program P is a finite set of rules. We denote by $\mathsf{data}(P)$ all the atoms constituting the ground facts of P.

Example 1. Let *P-Jungle* be a *Datalog*$^{\exists}$ program including the following rules:

r_1 : $\exists$`Z pursues(Z,X)` $\leftarrow$ `escapes(X)`
r_2 : `hungry(Y)` $\leftarrow$ `pursues(Y,X), fast(X)`
r_3 : `pursues(X,Y)` $\leftarrow$ `pursues(X,W), prey(Y)`
r_4 : `afraid(X)` $\leftarrow$ `pursues(Y,X), hungry(Y), strongerThan(Y,X).`

The program describes a funny scenario where an escaping, yet fast animal `X` may induce other animals to be afraid. Data for *P-Jungle* could be `escapes(gazelle)`, `fast(gazelle)`, `prey(antelope)`, `strongerThan(lion,antelope)`, and possibly also `pursues(lion,gazelle)`. We will use *P-Jungle* as a running example. □

Given a *Datalog*$^{\exists}$ program P, a *conjunctive query* (CQ) q over P is a first-order (FO) expression of the form $\exists \mathbf{Y}\, \mathbf{conj}_{[\mathbf{X} \cup \mathbf{Y}]}$, where $\mathbf{X}$ are its free variables. To highlight the free variables, we write $q(\mathbf{X})$ instead of q. Query q is a *Boolean CQ* (BCQ) if $\mathbf{X} = \emptyset$. Moreover, q is called *atomic* if $\mathbf{conj}$ is an atom.

Example 2. Animals pursed by a *lion* and stronger than some other animal can be retrieved by means of a CQ $\exists$`Y pursues(lion,X), strongerThan(X,Y).` □

Semantics and Query Answering. Given a set S of atoms and an atom $\mathbf{a}$, we say S entails $\mathbf{a}$ ($S \models \mathbf{a}$ for short) if there is a substitution σ s.t. $\sigma(\mathbf{a}) \in S$. Let $P \in$ *Datalog*$^{\exists}$. A set $M \subseteq \mathsf{base}(\Delta_C \cup \Delta_N)$ is a *model* for P ($M \models P$) if $M \models \sigma|_{\mathbf{X}}(\mathsf{head}(r))$ for each $r \in P$ of the form (1) and substitution σ s.t. $\sigma(\mathsf{body}(r)) \subseteq M$. Let $\mathsf{mods}(P)$ denote the set of models of P. Let $M \in \mathsf{mods}(P)$. A BCQ q is *true* w.r.t. M ($M \models q$) if there is a substitution σ s.t. $\sigma(\mathsf{atoms}(q)) \subseteq M$. Analogously, the answer of a CQ $q(\mathbf{X})$ w.r.t. M is the set $\mathsf{ans}(q, M) = \{\sigma|_{\mathbf{X}} : \sigma \text{ is a substitution } \wedge\ M \models \sigma|_{\mathbf{X}}(q)\}$. The answer of a CQ $q(\mathbf{X})$ w.r.t. a program P is the set $\mathsf{ans}_P(q) = \{\sigma : \sigma \in \mathsf{ans}(q, M)\ \forall M \in \mathsf{mods}(P)\}$. Note that for a BCQ q either $\mathsf{ans}_P(q) = \{\sigma|_\emptyset\}$ or $\mathsf{ans}_P(q) = \emptyset$. In the former case we say that q is (cautiously) *true* w.r.t. P, denoted by $P \models q$.

Let $\mathcal{C}$ be a class of *Datalog*$^{\exists}$ programs. *Query answering* (QA) over $\mathcal{C}$ is the following decision problem: *Given a program P in $\mathcal{C}$ and a BCQ q, determine whether $P \models q$ holds.* Class $\mathcal{C}$ is called QA-decidable if QA over $\mathcal{C}$ is decidable. We observe that computing $\mathsf{ans}_P(q)$ for a CQ $q(\mathbf{X})$ is Turing-reducible to QA. In fact, $\mathsf{ans}_P(q)$ is the set of substitutions σ s.t. the BCQ $\sigma(q)$ is true w.r.t. P. However, since $\sigma \in \mathsf{ans}_P(q)$ implies $\sigma(\mathbf{X}) \subseteq \mathsf{dom}(P)$, only finitely many substitutions have to be considered [13].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Let P be a *Datalog*$^{\exists}$ program. It is well-known that QA can be carried out by using a *universal model* of P, namely a model U of P s.t., for each BCQ q, $P \models q$ iff $U \models q$ [13]. A well-known procedure for constructing such a model U is the CHASE [17]. Intuitively, the CHASE first sets U to $\mathsf{data}(P)$. Next, it exhaustively repairs rules that are not satisfied in U by adding to U new atoms having "fresh" nulls in the positions of $\exists$-variables. Eventually, it terminates if $U \models P$. (See [19] for a formal definition.) Unfortunately, although the CHASE always constructs a (possibly infinite) universal model of P, QA remains undecidable in general [13].

3 *Shy*: A novel QA-decidable *Datalog*$^{\exists}$ class

The key idea behind this class intuitively relays on the following *shyness* property: *During a* CHASE*-run on a Shy program P, nulls propagated body-to-head do not meet each other to join.* In this regard, given a *Datalog*$^{\exists}$ program P, to detect whether a CHASE-run on P might produce some atom containing a null at a given position, we define the *null-set* of a position in an atom. More precisely, $\varphi^r_{\mathtt{X}}$ denotes the "representative" null introduced (during a CHASE-run on P) due to the $\exists$-variable $\mathtt{X}$ of the rule $r \in P$. (If $(r, \mathtt{X}) \neq (r', \mathtt{X}')$, then $\varphi^r_{\mathtt{X}} \neq \varphi^{r'}_{\mathtt{X}'}$.) Let $\mathbf{a}$ be an atom, and $\mathtt{X}$ a variable occurring in $\mathbf{a}$ at position i. The *null-set* of position i in $\mathbf{a}$ w.r.t. P, denoted by $\mathsf{nullset}(i, \mathbf{a})$, is inductively defined as follows. If $\mathbf{a}$ is the head atom of some rule $r \in P$, then $\mathsf{nullset}(i, \mathbf{a})$ is: (1) either the set $\{\varphi^r_{\mathtt{X}}\}$, if $\mathtt{X}$ is $\exists$-quantified in r; or (2) the intersection of every $\mathsf{nullset}(j, \mathbf{b})$ s.t. $\mathbf{b} \in \mathsf{body}(r)$ and $\mathtt{X}$ occurs at position j in $\mathbf{b}$, if $\mathtt{X}$ is $\forall$-quantified in r. If $\mathbf{a}$ is not a head atom, then $\mathsf{nullset}(i, \mathbf{a})$ is the union of $\mathsf{nullset}(i, \mathsf{head}(r))$ for each $r \in P$ s.t. $\mathsf{pred}(\mathsf{head}(r)) = \mathsf{pred}(\mathbf{a})$. A representative null φ *invades* a variable $\mathtt{X}$ that occurs at position i in an atom $\mathbf{a}$ if φ is contained in $\mathsf{nullset}(i, \mathbf{a})$. A variable $\mathtt{X}$ occurring in a conjunction of atoms **conj** is *attacked* in **conj** by a null φ if each occurrence of $\mathtt{X}$ in **conj** is invaded by φ. A variable $\mathtt{X}$ is *protected* in **conj** if it is attacked by no null. We are now ready to define the new *Datalog*$^{\exists}$ class.

Definition 1 ([19]). *Let Shy denote the class of all Datalog*$^{\exists}$ *programs containing only shy rules, where a rule r is called* shy *w.r.t. a program P if the following conditions are satisfied:* (i) *If a variable occurs in more than one body atom, then this variable is protected in $\mathsf{body}(r)$;* (ii) *If two distinct $\forall$-variables are not protected in $\mathsf{body}(r)$ but occur both in $\mathsf{head}(r)$ and in two different body atoms, then they are not attacked by the same null.* □

Resuming program *P-Jungle* of Example 1 we now prove its shyness. Let $\mathbf{a}_1, \ldots, \mathbf{a}_{12}$ be the atoms of rules r_1–r_4 in left-to-right/top-to-bottom order, and $\mathsf{nullset}(1, \mathbf{a}_1)$ be $\{\varphi^{r_1}_{\mathtt{Z}}\}$. First, we propagate $\varphi^{r_1}_{\mathtt{Z}}$ (head-to-body) to $\mathsf{nullset}(1, \mathbf{a}_4)$, $\mathsf{nullset}(1, \mathbf{a}_7)$, and $\mathsf{nullset}(1, \mathbf{a}_{10})$. Next, this null is propagated (body-to-head) from $\mathbf{a}_4$, $\mathbf{a}_7$ and $\mathbf{a}_3$ to $\mathsf{nullset}(1, \mathbf{a}_3)$, $\mathsf{nullset}(1, \mathbf{a}_6)$ and $\mathsf{nullset}(1, \mathbf{a}_{11})$, respectively. Finally, we observe that rules r_1–r_3 are trivially shy, and that r_4 also is because variable $\mathtt{Y}$ is not invaded in $\mathbf{a}_{12}$ even if $\varphi^{r_1}_{\mathtt{Z}}$ invades $\mathtt{Y}$ both in $\mathbf{a}_{10}$ and $\mathbf{a}_{11}$.

According to Definition 1, we observe that a program is *Shy* regardless its ground facts. Finally, we mention notable computational properties of this class.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Theorem 1 ([19]). *Checking whether a program P is shy is decidable. In particular, it is doable in polynomial-time.*

Theorem 2 ([19]). *QA over Shy is decidable. In particular, it is **P**-complete in data complexity for unrestricted conjunctive queries, and **EXP**-complete in combined complexity for atomic queries.*

4 Implementation and Experiments

We implemented a system, called DLV$^{\exists}$, that computes a very succinct set of atoms for answering CQs over *Shy* programs. Let P be a *Shy* program and q be a CQ. First, $\exists$-variables in rule heads are managed by skolemization. In particular, every rule $r \in P$ is skolemized, and skolemized terms are interpreted as functional symbols [8] within DLV$^{\exists}$. Next, the system singles out the set of predicates that are relevant for answering q by recursively traversing top-down (head-to-body) the rules in P, starting from the query predicates. This information is used to filter out, at loading time, the facts belonging to predicates irrelevant for answering the input query. At this point, the computation is further optimized rewriting P by a variant of the well-known Magic-Sets technique [11,3], that we adapted to *Datalog*$^{\exists}$.

We carried out an experimental analysis considering the well-known LUBM benchmark (see `http://swat.cse.lehigh.edu/projects/lubm/`). It refers to a university domain and includes a synthetic data generator, which we used to generate three increasing data sets, namely `lubm-10`, `lubm-30` and `lubm-50`. LUBM incorporates a set of 14 queries referred to as q_1–q_{14}. Tests were performed on an Intel Xeon X3430, 2.4 GHz, with 4 Gb Ram, running Linux Operating System. For each query, we allowed a maximum running time of 7200 seconds and a maximum memory usage of 2 Gb.

We compared DLV$^{\exists}$ with three state-of-the-art reasoners called Pellet [25], OWLIM-SE [4] and OWLIM-Lite [4]. Results are reported in Table 1, where

Table 1. Systems comparison: running time (sec.) and average time (G.Avg)

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{all}	G.Avg
lubm-10																
DLV$^{\exists}$	5	4	2	4	6	1	6	4	8	5	<1	1	6	2	17	2.87
Pellet	82	84	84	82	80	88	81	89	95	82	82	89	82	84	27	84.48
OWLIM-Lite	33	–	33	33	33	33	4909	70	–	33	33	33	33	33	33	53.31
OWLIM-SE	105	105	105	105	105	105	105	106	106	105	105	105	105	105	105	105.14
lubm-30																
DLV$^{\exists}$	16	13	7	14	21	3	21	12	25	18	<1	5	23	8	55	9.70
Pellet	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
OWLIM-Lite	107	–	107	106	107	106	–	528	–	107	106	106	107	106	106	123.18
OWLIM-SE	323	328	323	323	323	323	323	323	326	323	323	323	323	323	323	323.57
lubm-50																
DLV$^{\exists}$	27	23	12	23	35	6	34	22	42	31	<1	9	33	14	93	16.67
Pellet	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
OWLIM-Lite	188	–	190	187	189	188	–	1272	–	189	187	187	189	187	187	223.79
OWLIM-SE	536	547	536	536	536	537	536	536	542	536	536	536	536	537	536	537.35

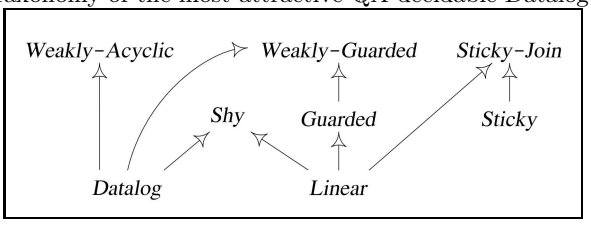
Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

times include the *total time* required for QA. We measured the total time, including data parsing and loading, because we are interested in ontology reasoning contexts where data and knowledge rapidly vary, even within hours. DLV$^{\exists}$ significantly outperforms all other systems in all tested queries and data sets. Comparing the other systems, OWLIM-Lite is in general faster than Pellet and OWLIM-SE. Pellet is faster than OWLIM-SE on `lubm-10`, but it answered no tested queries in the allotted time on `lubm-30` and `lubm-50`. Finally, column q_{all} shows the time taken by the systems to perform *fact inference*, namely to compute all atomic consequences required to answer any atomic query. Interestingly, even if DLV$^{\exists}$ is specifically designed for QA over frequently changing ontologies, it outperformed the competitors also in performing fact inference. Indeed, DLV$^{\exists}$ took about 17% and 51% of the time taken by OWLIM-SE and OWLIM-Lite.

5 Related Work and Discussion

Comparison with the literature reveals that *Shy* offers the best balance between expressivity and complexity among all known QA-decidable *Datalog*$^{\exists}$ classes relaying on the three main paradigms called *weak-acyclicity* [13], *guardness* [5] and *stickiness* [7]. Figure 1 provides a taxonomy of the most representative of these classes. In particular, for each pair $\mathcal{C}_1$ and $\mathcal{C}_2$ of classes, there is a direct path from $\mathcal{C}_1$ to $\mathcal{C}_2$ iff $\mathcal{C}_1 \subset \mathcal{C}_2$; also, $\mathcal{C}_1$ and $\mathcal{C}_2$ are not linked by any directed path iff they are uncomparable [19]. Table 2 summarizes the complexity of QA over these classes [19]. In both diagrams, only *Datalog* is intended to be $\exists$-free. In fact, among these classes, *Shy* is the only one supporting advanced, yet relevant properties such as role-transitivity and concept-product [24] (besides standard ontological properties such as role-hierarchy, role-inverse, concept-hierarchy). More specifically, even if *Weakly-Guarded* [5] encompasses and generalizes both *Datalog* and *Linear* [5] as *Shy*, it has untractable data complexity and no implementation. *Weakly-Acyclic* [13] and *Guarded* [5] are tractable (although they suffer of higher combined complexity than *Shy*) but the former does not include *Linear* (even the basic "father-person" ontology cannot be represented), while the latter does not include *Datalog* and supports neither role-transitivity nor concept-product. Moreover, no efficient implementation of *Guarded* has been found so far since the natural termination condition on *Guarded* programs requires the generation of a huge set of atoms for answering CQs. *Sticky* and *Sticky-Join* [7] are suitable for an efficient implementation and capture some light-weight DL properties but,

Fig. 1. Taxonomy of the most attractive QA-decidable *Datalog*$^{\exists}$ classes



Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

Class $\mathcal{C}$	*Data Complexity*	*Combined Complexity*
Weakly-Guarded	**EXP**-complete	**2EXP**-complete
Guarded, *Weakly-Acyclic*	**P**-complete	**2EXP**-complete
Datalog, *Shy*	**P**-complete	**EXP**-complete
Sticky, *Sticky-Join*	in $\mathbf{AC}_0$	**EXP**-complete
Linear	in $\mathbf{AC}_0$	**PSPACE**-complete

Table 2. Complexity of atomic query answering

since they do not generalize *Datalog*, they cannot express important KR features such as role-transitivity.

Regarding related systems, to the best of our knowledge, the only one directly supporting $\exists$-quantifiers in *Datalog* is Nyaya [12], which performs QA over *Linear-Datalog*$^{\exists}$. (We could not compare DLV$^{\exists}$ with Nyaya since it still provides no API for data loading and querying.) Concerning ontology reasoners, we mention QuOnto [2], Presto [23], Quest [21], Mastro [10] and OBDA [22] which rewrite axioms and queries to SQL. Such systems support standard FO semantics for unrestricted CQs, but the expressivity of their languages is limited to $\mathbf{AC}_0$ and excludes, e.g., transitivity property or concept products. The systems FaCT++ [26], RacerPro [15], Pellet [25] and HermiT [20] materialize all inferences at loading-time, implement very expressive description logics, but they do not support the standard FO semantics for CQs [14]. Actually, the Pellet system enables first-order CQs but only in the acyclic case. OWLIM [4] and KAON2 [16] perform full-materialization and implement expressive DLs, but they still miss to support the standard FO semantics for CQs [14].

Summing up, it turns out that DLV$^{\exists}$ is the first system supporting the standard FO semantics for unrestricted CQs with $\exists$-variables over ontologies using advanced properties (even beyond $\mathbf{AC}_0$), such as, role transitivity, role hierarchy, role inverse, and concept products. The experiments confirm the efficiency of DLV$^{\exists}$, which constitutes a powerful system for a fully-declarative QA.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases: The Logical Level*. Addison-Wesley Longman Publishing Co., Inc., 1995.
2. A. Acciarri, D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, and R. Rosati. QUONTO: querying ontologies. In *Proc. of the 20th AAAI Conf. on AI*, volume 4, pages 1670–1671, 2005.
3. M. Alviano, W. Faber, G. Greco, and N. Leone. Magic Sets for Disjunctive Datalog Programs. Technical Report 09/2009, Dept. of Math., Univ. of Calabria, IT, 2009. See `https://www.mat.unical.it/~faber/research/papers/TRMAT092009.pdf`.
4. B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, and R. Velkov. OWLIM: A family of scalable semantic repositories. *Semant. Web*, 2:33–42, 2011.
5. A. Calì, G. Gottlob, and M. Kifer. Taming the Infinite Chase: Query Answering under Expressive Relational Constraints. In *Proc. of the 11th KR Conf.*, pages 70–80, 2008. See: `http://dbai.tuwien.ac.at/staff/gottlob/CGK.pdf`.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

6. A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *Proc. of the 28th PODS Symp.*, pages 77–86, 2009.
7. A. Calì, G. Gottlob, and A. Pieris. Advanced Processing for Ontological Queries. *PVLDB*, 3(1):554–565, 2010.
8. F. Calimeri, S. Cozza, G. Ianni, and N. Leone. Enhancing ASP by Functions: Decidable Classes and Implementation Techniques. In *Proc. of the 24th AAAI Conf. on AI*, pages 1666–1670, 2010.
9. D. Calvanese, G. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *J. Autom. Reason.*, 39:385–429, 2007.
10. D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi, and D. F. Savo. The MASTRO system for ontology-based data access. *Semant. Web*, 2(1):43–53, 2011.
11. C. Cumbo, W. Faber, G. Greco, and N. Leone. Enhancing the Magic-Set Method for Disjunctive Datalog Programs. In *Proc. of the 20th ICLP*, volume 3132, pages 371–385, 2004.
12. R. De Virgilio, G. Orsi, L. Tanca, and R. Torlone. Semantic Data Markets: A Flexible Environment for Knowledge Management. In *Proc. of the 20th Int. CIKM*, pages 1559–1564, 2011.
13. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *TCS*, 336(1):89–124, 2005.
14. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic SHIQ. *JAIR*, 31(1):157–204, 2008.
15. V. Haarslev and R. Möller. RACER System Description. In *Proc. of the 6th IJCAR*, pages 701–705, 2001.
16. U. Hustadt, B. Motik, and U. Sattler. Reducing SHIQ- Descrption Logic to Disjunctive Datalog Programs. In *Proc. of the 9th KR Conf.*, pages 152–162, 2004.
17. D. Johnson and A. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *J. Comput. Syst. Sci.*, 28(1):167–189, 1984.
18. I. Kollia, B. Glimm, and I. Horrocks. SPARQL Query Answering over OWL Ontologies. In *Proc. of the 24th DL Int. Workshop*, volume 6643 of *LNCS*, pages 382–396. Springer, 2011.
19. N. Leone, M. Manna, G. Terracina, and P. Veltri. Efficiently Computable Datalog$^{\exists}$ Programs. In *Proc. of the 13th KR Int. Conf.*, 2012. To Appear. See `www.mat.unical.it/kr2012/`.
20. B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. *JAIR*, 36:165–228, 2009.
21. M. Rodriguez-Muro and D. Calvanese. Dependencies: Making Ontology Based Data Access Work in Practice. In *Proc. of the 5th AMW on Foundations of Data Management*, volume 477, 2011.
22. M. Rodriguez-Muro and D. Calvanese. Dependencies to Optimize Ontology Based Data Access. In *Description Logics*, volume 745. CEUR-WS.org, 2011.
23. R. Rosati and A. Almatelli. Improving Query Answering over DL-Lite Ontologies. In *Proc. of the 12th KR Conf.*, pages 290–300, 2010.
24. S. Rudolph, M. Krötzsch, and P. Hitzler. All Elephants are Bigger than All Mice. In *Proc. of the 21st DL Int. Workshop*, volume 353, 2008.
25. E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semant.*, 5(2):51–53, 2007.
26. D. Tsarkov and I. Horrocks. FaCT++ Description Logic Reasoner: System Description. In *Proc. of the 3rd IJCAR*, volume 4130, pages 292–297, 2006.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

The Algebra and the Logic for SQL Nulls

Enrico Franconi and Sergio Tessaris

Free University of Bozen-Bolzano, Italy
lastname@inf.unibz.it

Abstract The logic of nulls in databases has been subject of investigation since their introduction in Codd's Relational Model, which is the foundation of the SQL standard. In the logic based approaches to modelling relational databases proposed so far, nulls are considered as representing unknown values. Such existential semantics fails to capture the behaviour of the SQL standard. We show that, according to Codd's Relational Model, a SQL null value represents a non-existing value; as a consequence no indeterminacy is introduced by SQL null values. We show that the domain independent fragment of the extension of first-order logic accounting for predicates with missing arguments is equivalent to Codd's relational algebra with SQL nulls. Moreover, we illustrate a faithful encoding of the logic into standard first-order logic. At the end, we show how to capture in this framework the UNIQUE, PRIMARY KEY, and FOREIGN KEY constraints as defined in the SQL:1999 standard.

1 Relational Databases and SQL Null Values

Consider a database instance with null values over the relational schema $\{R/2\}$, and an SQL query asking for the tuples in R being equal to themselves:

R :

1	2
a	b
b	**N**

```
SELECT * FROM R
WHERE R.1 = R.1 AND R.2 = R.2 ;
```

$\Longrightarrow$

```
 1 | 2
---+---
 a | b
(1 row)
```

Figure 1.

In SQL, the query above returns the table `R` if and only if the table `R` does not have any null value, otherwise it returns just the tuples not containing a null value, i.e., in this case only the first tuple $\langle a, b\rangle$. Informally this is due to the fact that an SQL null value is never equal (or not equal) to anything, including itself. How can we formally capture this behaviour?

We introduce a formal semantics for SQL null values in order to capture exactly the behaviour of SQL queries and SQL constraints in presence of null values. We restrict our attention to the first-order fragment of SQL – e.g., we do not consider aggregates, and both queries and constraints should be set based (as opposed to bag/multi-set based): this fragment can be expressed, in absence of null values, into the standard relational algebra $\mathcal{RA}$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

To the best of our knowledge, there has been no attempt to formalise in logic a relational algebra with *SQL* null values. It is well known that SQL null values require a special semantics. Indeed, if they were treated as standard database constants, the direct translation in the standard relational algebra $\mathcal{RA}$ of the above SQL query would be equivalent to the *identity* expression for R: $\sigma_{1=1}$ $\sigma_{2=2}$ R, giving as an answer the table R itself, namely the tuples $\{\langle a, b\rangle, \langle b, \mathbf{N}\rangle\}$. However, we have seen that the expected answer to this query is different.

The most popular semantics in the literature for null values is the one interpreting null values with an existential meaning, namely a null value denotes an object which exists but has an unknown identity: this is the semantics of *naive* tables (see [1]). It is easy to see that also with this semantics, the direct translation of the above SQL query in the standard relational algebra $\mathcal{RA}$ over naive tables would be equivalent to the *identity* expression for R: $\sigma_{1=1}$ $\sigma_{2=2}$ R, giving as an answer the table R itself, namely the tuples $\{\langle a, b\rangle, \langle b, \mathbf{N}\rangle\}$. And again, we have seen above that the expected answer to this query is different.

2 Database Instances with Null Values

We introduce the notions of tuple, relation, and database instance in presence of null values. We consider the unnamed (positional) perspective for the attributes of tuples: elements of a tuple are identified by their position within the tuple.

Given a set of domain values Δ, an n-tuple is a *total* function from integers from 1 up to n (the position of the element within the tuple) into the set of domain values Δ augmented by the special term $\mathbf{N}$ (the null value): if we denote the set $\{i \in \mathbb{N} \mid 1 \leq i \wedge i \leq n\}$, possibly empty if $n = 0$, as $[1 \cdots n]$, then an n-tuple is a total function $[1 \cdots n] \mapsto \Delta \cup \{\mathbf{N}\}$. Note that the zero-tuple is represented by a *constant* zero-ary function that we call $\{\}$. For example, the tuple $\langle b, \mathbf{N}\rangle$ is represented as $\{1 \mapsto b, 2 \mapsto \mathbf{N}\}$, while the zero-tuple $\langle\rangle$ is represented as $\{\}$. A relation of arity n is a set of n-tuples; if we want to specify that n is the arity of a given relation R, we write the relation as R/n. Note that a relation of arity zero is either empty or it includes only the zero-tuple $\{\}$. A relational schema $\mathcal{R}$ includes a set of relation symbols with their arities and a set of constant symbols $\mathcal{C}$. A database instance $\mathcal{I}^{\mathbf{N}}$ associates to each relation symbol R of arity n from the relational schema $\mathcal{R}$ a set of n-tuples $\mathcal{I}^{\mathbf{N}}(R)$, and to each constant symbol in $\mathcal{C}$ a domain value in Δ. Usually, in relational databases all constant symbols are among the domain values and are associated in the database instance to themselves – this is called the Standard Name Assumption. As an example of a database instance $\mathcal{I}^{\mathbf{N}}$ consider Figure 2(a).

In our work we consider two alternative representations of a database instance $\mathcal{I}^{\mathbf{N}}$, where null values do not appear explicitly: $\mathcal{I}^{\varepsilon}$ and $\mathcal{I}^{\wp}$. We will show that they are isomorphic to $\mathcal{I}^{\mathbf{N}}$. The representations share the same constant symbols, domain values, and mappings from constant symbols to domain values. The differences are in the way null values are encoded within a tuple.

Compared with a database instance $\mathcal{I}^{\mathbf{N}}$, a corresponding database instance $\mathcal{I}^{\varepsilon}$ differs only in the way it represents n-tuples: an n-tuple is a *partial* function

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

$R/2:\ \{\{1 \mapsto a, 2 \mapsto b\},\ \{1 \mapsto b, 2 \mapsto \mathbf{N}\}\}$

(a) instance $\mathcal{I}^{\mathbf{N}}$

$R/2:\ \{\{1 \mapsto a, 2 \mapsto b\},\ \{1 \mapsto b\}\}$

(b) instance $\mathcal{I}^{\varepsilon}$

$\widetilde{R}_{\{1,2\}}/2 : \{\{1 \mapsto a, 2 \mapsto b\}\}$

$\widetilde{R}_{\{1\}}/1 : \{\{1 \mapsto b\}\}$

$\widetilde{R}_{\{2\}}/1 : \{\}$

$\widetilde{R}_{\{\}}/0 : \{\}$

(c) instance $\mathcal{I}^{\wp}$

Figure 2. The three representations of the database instance of Figure 1

from integers from 1 up to n into the set of domain values - the function is undefined exactly for those positional arguments which are otherwise defined and mapped to a null value in $\mathcal{I}^{\mathbf{N}}$. As an example of a database instance $\mathcal{I}^{\varepsilon}$ consider Figure 2(b).

Compared with a database instance $\mathcal{I}^{\mathbf{N}}$, a corresponding database instance $\mathcal{I}^{\wp}$ differs in the way relation symbols are interpreted: a relation symbol of arity n is associated to a set of null-free tuples of dishomogeneous arities up to n. Given a database instance $\mathcal{I}^{\mathbf{N}}$ defined over a relational schema $\mathcal{R}$, the corresponding database instance $\mathcal{I}^{\wp}$ is defined over the *decomposed* relational schema $\widetilde{\mathcal{R}}$: for each relation symbol $R \in \mathcal{R}$ of arity n and for each (possibly empty) subset of its positional arguments $A \subseteq [1 \cdots n]$, the decomposed relational schema $\widetilde{\mathcal{R}}$ includes a predicate $\widetilde{R}_A$ with arity $|A|$. The correspondence between $\mathcal{I}^{\mathbf{N}}$ and $\mathcal{I}^{\wp}$ is based on the fact that each $|A|$-tuple in the relation $\mathcal{I}^{\wp}(\widetilde{R}_A)$ corresponds exactly to the n-tuple in $\mathcal{I}^{\mathbf{N}}(R)$ having non-null values only in the positional arguments in A, with the same values and in the same relative positional order. This corresponds to the notion of lossless *horizontal decomposition* in databases [2]. As an example of a database instance $\mathcal{I}^{\wp}$ consider Figure 2(c).

In absence of null values, the $\mathcal{I}^{\mathbf{N}}$ and $\mathcal{I}^{\varepsilon}$ representations of a database instance coincide, and they coincide also with the $\mathcal{I}^{\wp}$ representation if we equate each n-ary R relation symbol in $\mathcal{I}^{\mathbf{N}}$ and $\mathcal{I}^{\varepsilon}$ with its corresponding $\widetilde{R}_{[1\cdots n]}$ relation symbol in $\mathcal{I}^{\wp}$ – indeed in absence of null values $\mathcal{I}^{\mathbf{N}}(R) = \mathcal{I}^{\varepsilon}(R) = \mathcal{I}^{\wp}(\widetilde{R}_{[1\cdots n]})$ for every n-ary relation R, and $\mathcal{I}^{\wp}(\widetilde{R}_A) = \emptyset$ for every relation of arity $|A| < n$. Given the discussed isomorphisms, in the following – whenever the difference in the representation of null values is not ambiguous – we will denote as $\mathcal{I}$ the database instance represented in any of the above three forms $\mathcal{I}^{\mathbf{N}}$, $\mathcal{I}^{\varepsilon}$, or $\mathcal{I}^{\wp}$.

3 Relational Algebra with Null Values

We introduce in this Section the formal semantics of the relational algebra dealing with null values, corresponding (modulo the zero-ary relations) to the first-order fragment of SQL.

Let's first recall the notation of the standard relational algebra $\mathcal{RA}$ (see, e.g., [3] for details). Standard relational algebra expressions over a relational schema $\mathcal{R}$ are built according to the inductive formation rules in the boxed expressions of

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Atomic relation - R **-** (where $R \in \mathcal{R}$)

$$R(\mathcal{I}) = \mathcal{I}^{\mathbf{N}}(R).$$

Constant singleton - $\langle v \rangle$ **-** (where $v \in \mathcal{C}$)

$$\langle v \rangle(\mathcal{I}) = \{1 \mapsto v\}.$$

Selection - $\sigma_{i=v}\ e$, $\sigma_{i=j}\ e$ **-** (where $v \in \mathcal{C}$, ℓ is the arity of e, and $i, j \leq \ell$)

$$\sigma_{i=v}\ e(\mathcal{I}) = \{s \text{ is a } \ell\text{-tuple} \mid s \in e(\mathcal{I}) \wedge s(i) = v\},$$
$$\sigma_{i=j}\ e(\mathcal{I}) = \{s \text{ is a } \ell\text{-tuple} \mid s \in e(\mathcal{I}) \wedge s(i) = s(j)\}.$$

Projection - $\pi_{i_1,\ldots,i_k} e$ **-** (where ℓ is the arity of e, and $\{i_1, \ldots, i_k\} \subseteq [1 \ldots \ell]$)

$$\pi_{i_1,\ldots,i_k} e\ (\mathcal{I}) = \{s \text{ is a } k\text{-tuple} \mid \text{exists } s' \in e(\mathcal{I}) \text{ s.t. for all } 1 \leq j \leq k.\ s(j) = s'(i_j)\}.$$

Cartesian product - $e \times e'$ **-** (where n, m are the arities of e, e')

$$(e \times e')(\mathcal{I}) = \{s \text{ is a } (n+m)\text{-tuple} \mid \text{exists } t \in e(\mathcal{I}), t' \in e'(\mathcal{I}) \text{ s.t.}$$
$$\text{for all } \quad 1 \leq j \leq n. \quad s(j) = t(j) \wedge$$
$$\text{for all } 1 + n \leq j \leq (n+m).\ s(j) = t'(j-n)\}.$$

Union/Difference - $e \cup e'$, $e - e'$ **-** (where ℓ is the arity of e and e')

$$(e \cup e')(\mathcal{I}) = \{s \text{ is a } \ell\text{-tuple} \mid s \in e(\mathcal{I}) \vee s \in e'(\mathcal{I})\},$$
$$(e - e')(\mathcal{I}) = \{s \text{ is a } \ell\text{-tuple} \mid s \in e(\mathcal{I}) \wedge s \notin e'(\mathcal{I})\}.$$

Derived operators - where $v \in \mathcal{C}$, $\ell \leq \mathsf{min}(m, n)$, m, n are the arities of e, e', $i, j, i_1, \ldots, i_\ell \leq m$, and $k_1, \ldots, k_\ell \leq n$

$$\sigma_{i<>v}\ e \equiv e - \sigma_{i=v}\ e,$$
$$\sigma_{i<>j}\ e \equiv e - \sigma_{i=j}\ e,$$
$$e \underset{\substack{i_1=k_1 \\ \cdots \\ i_\ell=k_\ell}}{\bowtie} e' \equiv \pi_{([1\cdots m+n] \setminus \{m+k_1,\ldots,m+k_\ell\})} \sigma_{i_1=m+k_1} \ldots \sigma_{i_\ell=m+k_\ell}\ (e \times e')\ .$$

Figure 3. The standard relational algebra $\mathcal{RA}$

Figure 3. Also in Figure 3 the semantics of an algebra expression e is inductively defined as the transformation of database instances $\mathcal{I}$ – with the Standard Name Assumption – to a set of tuples $e(\mathcal{I})$.

We extend the standard relational algebra to deal with SQL nulls; we refer to it as *Null Relational Algebra* ($\mathcal{RA}^{\mathbf{N}}$). In order to deal with null values in the relational model, Codd [4] included the special null value in the domain and adopted a three-valued logic having the third truth value *unknown* together with *true* and *false*. The comparison expressions in Codd's algebra are evaluated to *unknown* if they involve a null value, while in set operations, tuples otherwise identical but containing null values are considered to be different. SQL uses this three-valued logic for the evaluation of `WHERE` clauses [5]. In order to define

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Null singleton - $\langle \mathbf{N} \rangle$ **-**

$$\langle \mathbf{N} \rangle(\mathcal{I}) = \{1 \mapsto \mathbf{N}\}.$$

Selection - $\sigma_{i=j}\ e$ **-** (where ℓ is the arity of e, and $i, j \leq \ell$)

$$\sigma_{i=j}\ e(\mathcal{I}) = \{s \text{ is a } \ell\text{-tuple} \mid\ s \in e(\mathcal{I}) \wedge s(i) = s(j) \wedge s(i) \neq \mathbf{N} \wedge s(j) \neq \mathbf{N}\}.$$

Derived operators - where $v \in \mathcal{C}$, $\ell \leq \mathsf{min}(m, n)$, m, n are the arities of e, e', $i, j, i_1, \ldots, i_\ell \leq m$, and $k_1, \ldots, k_\ell \leq n$

$$\begin{aligned}
\sigma_{i<>j}\ e &\equiv \sigma_{i=i}\ \sigma_{j=j}\ e - \sigma_{i=j}\ e,\\
\sigma_{\mathsf{isNull}(i)}\ e &\equiv e - \sigma_{i=i}\ e,\\
\sigma_{\mathsf{isNotNull}(i)}\ e &\equiv \sigma_{i=i}\ e,\\
\sigma_{i=\mathbf{N}}\ e &\equiv e - e,\\
\sigma_{i<>\mathbf{N}}\ e &\equiv e - e,\\
e \underset{\substack{i_1=k_1\\ \cdots\\ i_\ell=k_\ell}}{\rtimes} e' &\equiv (e \underset{\substack{i_1=k_1\\ \cdots\\ i_\ell=k_\ell}}{\bowtie} e') \cup (e - \pi_{1,\ldots,m}(e \underset{\substack{i_1=k_1\\ \cdots\\ i_\ell=k_\ell}}{\bowtie} e')) \times (\underbrace{\langle \mathbf{N} \rangle \times \cdots \times \langle \mathbf{N} \rangle}_{n-\ell}).
\end{aligned}$$

Figure 4. The null relational algebra $\mathcal{RA}^{\mathbf{N}}$ defined only in the parts different from $\mathcal{RA}$

$\mathcal{RA}^{\mathbf{N}}$ from the standard relational algebra, we adopt the $\mathcal{I}^{\mathbf{N}}$ representation of a database instance where null values are explicitly present as possible elements of tuples, and with the Standard Name Assumption. All the $\mathcal{RA}$ expressions are valid $\mathcal{RA}^{\mathbf{N}}$ expressions, and maintain the same semantics, with the only change in the semantics of the *selection* expressions $\sigma_{\mathbf{i=j}}\ \mathbf{e}$ and $\sigma_{\mathbf{i<>j}}\ \mathbf{e}$ with equality or inequality: in these cases the semantic definitions make sure that the elements to be tested for equality (or inequality) are both different from the null value in order for the equality (or inequality) to succeed. In other words, it is enough to let equality (and inequality) fail whenever null values are involved, since its evaluation would be *unknown*.

The syntax of $\mathcal{RA}^{\mathbf{N}}$ expressions extends the standard $\mathcal{RA}$ syntax only for the additional *null singleton* expression $\langle \mathbf{N} \rangle$ (and for the derived operators involving null values). Figure 4 introduces the syntax and semantics of $\mathcal{RA}^{\mathbf{N}}$ expressed in terms of its difference with $\mathcal{RA}$, including the derived operators which are added or defined differently in $\mathcal{RA}^{\mathbf{N}}$. It is easy to show that the definition of the null relational algebra exactly matches the (informal) definition given to SQL with null values, and it generates the same practical behaviour [5].

Given a tuple t and a $\mathcal{RA}^{\mathbf{N}}$ expression e, we call *models* of the expression e with the answer t all the database instances $\mathcal{I}$ such that $t \in e(\mathcal{I})$. Sometimes an n-ary algebra expression is intended to express a boolean statement over a database instance in the form of a *denial constraint*: this is done by checking whether its evaluation over the database instance returns the empty n-ary relation or not. an n-ary denial constraint e can always be reduced into a zero-ary constraint, whose boolean value is meant to be **true** if its evaluation contains the zero-tuple $\{\}$ and **false** if it is empty, by considering its zero-ary projection $(\pi_\emptyset e)$.

Example 1. Consider the schema $\{R/2, S/2\}$ with the following data and $\mathcal{RA}^{\mathbf{N}}$ constraints:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

R :

1	2
a	b
b	**N**

S :

1	2
a	a
a	**N**
N	a
N	**N**

- UNIQUE constraint for $R.1$: $\sigma_{1=3}\ \sigma_{2\neq 4}\ (R \times R) = \emptyset$;
- NOT-NULL constraint for $R.1$: $\sigma_{\mathsf{isNull}(1)}\ R = \emptyset$;
- UNIQUE constraint for $S.1$: $\sigma_{1=3}\ \sigma_{2\neq 4}\ (S \times S) = \emptyset$,
- FOREIGN KEY constraint from $S.2$ to $R.1$: $\pi_2\sigma_{\mathsf{isNotNull}(2)}\ S - \pi_1\sigma_{\mathsf{isNotNull}(1)}\ R = \emptyset$.

It is easy to see that these constraints are all satisfied: they behave in the same way as the corresponding SQL constraints with null values. Similarly, the query considered in the previous Section ($\sigma_{1=1}\ \sigma_{2=2}\ R$), now behaves as in SQL and it returns correctly $\{\langle a, b\rangle\}$.

4 First-order Logic with Null Values

The *Null Relational Calculus* ($\mathcal{FOL}^{\varepsilon}$) is a first-order logic language with an explicit symbol **N** representing the null value, and where predicates denote tuples over *subsets* of the arguments instead of just their whole set of arguments. It extends classical first-order logic in order to take into account the possibility that some of the arguments of a relation might not exist.

Given a set of predicate symbols each one associated with an arity together with the special equality binary predicate $=$, and a set $\mathcal{C}$ of constants – together forming the relational schema (or *signature*) $\mathcal{R}$ – and a set of variable symbols, terms of $\mathcal{FOL}^{\varepsilon}$ are variables, constants, and the special *null* symbol **N**, and formulae of $\mathcal{FOL}^{\varepsilon}$ are defined by the following rules:

1. if $t_1, \ldots, t_n$ are terms and R is a predicate symbol in $\mathcal{R}$ (different from the equality) of arity n, $R(t_1, \ldots, t_n)$ is an atomic formula;
2. if t_1, t_2 are terms different from **N**, $=(t_1, t_2)$ is an atomic formula;
3. if φ and φ' are formulae, then $\neg\varphi$, $\varphi \wedge \varphi'$ and $\varphi \vee \varphi'$ are formulae;
4. if φ is a formula and x is a variable, then $\exists x\varphi$ and $\forall x\varphi$ are formulae.

The semantics of $\mathcal{FOL}^{\varepsilon}$ formulae is given in terms of database instances of type $\mathcal{I}^{\varepsilon}$, called *interpretations*. As usual, an interpretation $\mathcal{I}^{\varepsilon}$ includes an interpretation domain Δ and it associates each relation symbol R of arity n in the signature to a set of n-tuples $\mathcal{I}^{\varepsilon}(R)$ – i.e., a set of *partial* functions with range in Δ – and each constant symbol in $\mathcal{C}$ to a domain value in Δ (we do not consider here the Standard Name Assumption). The equality predicate is interpreted as the classical equality over the domain Δ. It is easy to see that if n-tuples were just *total* functions, then an interpretation $\mathcal{I}^{\varepsilon}$ would correspond exactly to a classical first-order interpretation.

The definition of satisfaction and entailment in $\mathcal{FOL}^{\varepsilon}$ is the same as in classical first-order logic with equality over the signature $\mathcal{R}$, with the only difference in the truth value of atomic formulae which use partial functions instead of total functions, because of the possible presence of null values. As usual, an interpretation $\mathcal{I}^{\varepsilon}$ satisfying a formula is called a *model* of the formula.
An interpretation $\mathcal{I}^{\varepsilon}$ and a valuation function for variable symbols α satisfy

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

an atomic formula – written $\mathcal{I}^{\varepsilon}, \alpha \models_{\mathcal{FOL}^{\varepsilon}} R(t_1, \ldots, t_n)$ – iff there is an n-tuple $\tau \in \mathcal{I}^{\varepsilon}(R)$ such that for each $i \in [1 \cdots n]$: $\tau(i) = \mathcal{I}^{\varepsilon}(c)$ if t_i is a constant symbol c, $\tau(i) = \alpha(t_i)$ if t_i is a variable symbol, and $\tau(i)$ is undefined if $t_i = \mathbf{N}$. It is easy to see that the satisfiability of a $\mathcal{FOL}^{\varepsilon}$ formula without any occurrence of the null symbol $\mathbf{N}$ doesn't depend on partial tuples; so its models can be characterised by classical first-order semantics: in each model the interpretation of predicates would include only tuples represented as total functions.

Example 2. The models of the $\mathcal{FOL}^{\varepsilon}$ formula $R(a, b) \wedge R(b, \mathbf{N})$ are the interpretations $\mathcal{I}^{\varepsilon}$ such that $\mathcal{I}^{\varepsilon}(R)$ includes the tuples $\{1 \mapsto a, 2 \mapsto b\}$ and $\{1 \mapsto b\}$.

4.1 Characterisation in classical First-order Logic

Given a signature $\mathcal{R}$, let's consider a classical first-order logic language with equality ($\mathcal{FOL}$) over the *decomposed* signature $\widetilde{\mathcal{R}}$, as it has been defined in Section 2; $\mathcal{FOL}$ has a classical semantics with models of type $\mathcal{I}^{\wp}$ and it does not deal with null values directly. In this Section we show that $\mathcal{FOL}^{\varepsilon}$ over $\mathcal{R}$ and $\mathcal{FOL}$ over $\widetilde{\mathcal{R}}$ are equally expressive, namely that for every formula in $\mathcal{FOL}^{\varepsilon}$ over the signature $\mathcal{R}$ there is a corresponding formula in $\mathcal{FOL}$ over the decomposed signature $\widetilde{\mathcal{R}}$, such that the two formulae have isomorphic models, and that for every formula in $\mathcal{FOL}$ over the decomposed signature $\widetilde{\mathcal{R}}$ there is a corresponding formula in $\mathcal{FOL}^{\varepsilon}$ over the signature $\mathcal{R}$, such that the two formulae have isomorphic models. As we discussed before, the isomorphism between the interpretations $\mathcal{I}^{\mathbf{N}}$ and $\mathcal{I}^{\wp}$ is based on the fact that each $|A|$-tuple in the relation $\mathcal{I}^{\wp}(\widetilde{R}_A)$ corresponds exactly to the n-tuple in $\mathcal{I}^{\mathbf{N}}(R)$ having non-null values *only* in the positional arguments specified in A, with the same values and in the same relative positional order.

In order to relate the two logics, we define a bijective translation function $\Omega_f(\cdot)$ (and its inverse $\Omega_f^{-1}(\cdot)$) which maps $\mathcal{FOL}^{\varepsilon}$ formulae into $\mathcal{FOL}$ formulae (and vice-versa).

Definition 1 (Bijective translation Ω_f).
$\Omega_f(\cdot)$ is a bijective function from $\mathcal{FOL}^{\varepsilon}$ formulae over the signature $\mathcal{R}$ to $\mathcal{FOL}$ formulae over the signature $\widetilde{\mathcal{R}}$, defined as follows.

- $\Omega_f(R(t_1, \ldots, t_n)) = \widetilde{R}_{\{i_1,\ldots,i_k\}}(t_{i_1}, \ldots, t_{i_k})$,
  where $R \in \mathcal{R}$ an n-ary relation, $\{i_1, \ldots, i_k\} = \{j \in [1 \cdots n] \mid t_j$ is not $\mathbf{N}\}$.
  Obviously: $\Omega_f^{-1}(\widetilde{R}_{\{i_1,\ldots,i_k\}}(t_{i_1}, \ldots, t_{i_k})) = R(t'_1, \ldots, t'_n)$,
  where $\{i_1, \ldots, i_k\} \subseteq [1 \cdots n]$ and $t_{i_j} = t'_j$ for $j = 1, \ldots, k$, and t'_j is $\mathbf{N}$ for $j \in [1 \cdots n] \setminus \{i_1, \ldots, i_k\}$.
  In both the direct and inverse cases we assume $i_1, \ldots, i_k$ in ascending order.
- *The translation of equality atoms and non atomic formulae is the identity transformation inductively defined on top of the above translation of atomic formulae.*

Example 3. The $\mathcal{FOL}^{\varepsilon}$ formula $\exists x.R(a, x) \wedge R(x, \mathbf{N}) \wedge R(\mathbf{N}, \mathbf{N})$ over the signature R is translated as the $\mathcal{FOL}$ formula $\exists x.\widetilde{R}_{\{1,2\}}(a, x) \wedge \widetilde{R}_{\{1\}}(x) \wedge \widetilde{R}_{\{\}}$ over the decomposed signature $\widetilde{\mathcal{R}}$, and vice-versa.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

The above bijective translation preserves the models of the formulae, modulo the isomorphism among models presented in Section 2.

Theorem 1. *Let φ be a $\mathcal{FOL}^{\varepsilon}$ formula over the signature $\mathcal{R}$, and $\widetilde{\varphi}$ a $\mathcal{FOL}$ formula over the signature $\widetilde{\mathcal{R}}$. Then for any (database) instance $\mathcal{I}$:*

$$\mathcal{I}^{\varepsilon}, \alpha \models_{\mathcal{FOL}^{\varepsilon}} \varphi \quad \textit{if and only if} \quad \mathcal{I}^{\wp}, \alpha \models_{\mathcal{FOL}} \Omega_f(\varphi).$$

4.2 Domain Independent fragment of $\mathcal{FOL}^{\varepsilon}$ with Standard Names

In this Section we introduce the domain independent fragment of $\mathcal{FOL}^{\varepsilon}$ with the Standard Name Assumption, and we analyse its properties.

Definition 2 (Domain Independence). *A $\mathcal{FOL}^{\varepsilon}$ closed formula φ is domain independent if for every two interpretations $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \mathcal{I}(\cdot) \rangle$ and $\mathcal{J} = \langle \Delta^{\mathcal{J}}, \mathcal{J}(\cdot) \rangle$, which agree on the interpretation of relation symbols and constant symbols – i.e. $\mathcal{I}(\cdot) = \mathcal{J}(\cdot)$ – but disagree on the interpretation domains $\Delta^{\mathcal{I}}$ and $\Delta^{\mathcal{J}}$:*

$$\mathcal{I} \models \varphi \quad \textit{if and only if} \quad \mathcal{J} \models \varphi.$$

The domain independent fragment of $\mathcal{FOL}^{\varepsilon}$ includes only the $\mathcal{FOL}^{\varepsilon}$ domain independent formulae.

It is easy to see that the domain independent fragment of $\mathcal{FOL}^{\varepsilon}$ can be characterised with the *safe-range* syntactic fragment of $\mathcal{FOL}^{\varepsilon}$: intuitively, a formula is safe-range if and only if its variables are bounded by positive predicates or equalities – for the syntactical definition see, e.g., [3]. Due to the strong semantic equivalence expressed in Theorem 1 and to the fact the the bijection $\Omega_f(\cdot)$ preserves the syntactic structure of the formulae, we can reuse the results about safe-range transformations and domain independence holding for classical $\mathcal{FOL}$.

Theorem 2. *Any safe-range $\mathcal{FOL}^{\varepsilon}$ formula is domain independent, and any domain independent $\mathcal{FOL}^{\varepsilon}$ formula can be transformed into a logically equivalent safe-range $\mathcal{FOL}^{\varepsilon}$ formula.*

We observe that an interpretation is a model of a formula in the domain independent fragment of $\mathcal{FOL}^{\varepsilon}$ with the Standard Name Assumption *if and only if* the interpretation which agrees on the interpretation of relation and constant symbols but with the interpretation domain equal to the set of standard names $\mathcal{C}$ is a model of the formula. Therefore, in the following when dealing with the domain independent fragment of $\mathcal{FOL}^{\varepsilon}$ with the Standard Name Assumption we can just consider interpretations with the interpretation domain equal to $\mathcal{C}$.

We can weaken the Standard Name Assumption by assuming *Unique Names* instead. An interpretation $\mathcal{I}$ satisfies the Unique Name Assumption if $\mathcal{I}(a) \neq \mathcal{I}(b)$ for any different $a, b \in \mathcal{C}$. An interpretation is a model of a $\mathcal{FOL}^{\varepsilon}$ formula with the Standard Name Assumption *if and only if* the interpretation obtained by homomorphically transform the standard names with arbitrary domain elements is a model of the formula; this latter interpretation satisfies the Unique

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Name Assumption. It is possible therefore to interchange the Standard Name and the Unique Name Assumptions; this is of practical advantage, since the Unique Name Assumption can be encoded in first-order logic and it is natively present in most description logic reasoners.

5 Equivalence of Algebra and Calculus

We show that the $\mathcal{RA}^{\mathbf{N}}$ relational algebra with nulls and the domain independent fragment of $\mathcal{FOL}^{\varepsilon}$ with the Standard Name Assumption are equally expressive. The notion of *equal expressivity* is captured by the following two theorems.

Theorem 3. *Let e be an arbitrary $\mathcal{RA}^{\mathbf{N}}$ expression of arity n, and t an arbitrary n-tuple as a total function with values taken from the set $\mathcal{C} \cup \{\mathbf{N}\}$. There is a function $\Omega(e,t)$ translating e with respect to t into a closed safe-range $\mathcal{FOL}^{\varepsilon}$ formula, such that for any instance $\mathcal{I}$ with the Standard Name Assumption:*

$$t \in e(\mathcal{I}^{\mathbf{N}}) \quad \textit{if and only if} \quad \mathcal{I}^{\varepsilon} \models_{\mathcal{FOL}^{\varepsilon}} \Omega(e,t).$$

Theorem 4. *Let φ be an arbitrary safe-range $\mathcal{FOL}^{\varepsilon}$ closed formula. There is a $\mathcal{RA}^{\mathbf{N}}$ expression e, such that for any instance $\mathcal{I}$ with the Standard Name Assumption:*

$$\mathcal{I}^{\varepsilon} \models_{\mathcal{FOL}^{\varepsilon}} \varphi \quad \textit{if and only if} \quad e(\mathcal{I}^{\mathbf{N}}) \neq \emptyset.$$

This means that there exists a reduction from the membership problem of a tuple in the answer of a $\mathcal{RA}^{\mathbf{N}}$ expression over a database instance with null values into the satisfiability problem of a closed safe-range $\mathcal{FOL}^{\varepsilon}$ formula over the same database (modulo the isomorphism among database instances presented in Section 2); and there exists a reduction from the satisfiability problem of a closed safe-range $\mathcal{FOL}^{\varepsilon}$ formula over a database instance with null values into the emptiness problem of the answer of a $\mathcal{RA}^{\mathbf{N}}$ expression over the same database (modulo the isomorphism among database instances).

This is the translation mapping $\mathcal{RA}^{\mathbf{N}}$ expressions into safe-range $\mathcal{FOL}^{\varepsilon}$ formulae.

Definition 3 (From $\mathcal{RA}^{\mathbf{N}}$ to safe-range $\mathcal{FOL}^{\varepsilon}$). *Let e be an arbitrary $\mathcal{RA}^{\mathbf{N}}$ expression, and t an arbitrary tuple of the same arity as e, as a total function with values taken from the set $\mathcal{C} \cup \{\mathbf{N}\} \cup \mathcal{V}$, where $\mathcal{V}$ is a countable set of variable symbols. The function $\Omega(e,t)$ translates e with respect to t into a $\mathcal{FOL}^{\varepsilon}$ formula according to the following inductive definition:*

- *for any $R/_{\ell} \in \mathcal{R}$, $\Omega(R,t) \rightsquigarrow R(t(1),\ldots,t(\ell))$*
- $\Omega(\langle v \rangle, t) \rightsquigarrow \begin{cases} =(t(1), v) & \textit{if } t(1) \neq \mathbf{N} \\ \mathbf{false} & \textit{otherwise} \end{cases}$
- $\Omega(\langle \mathbf{N} \rangle, t) \rightsquigarrow \begin{cases} \mathbf{false} & \textit{if } t(1) \neq \mathbf{N} \\ \mathbf{true} & \textit{otherwise} \end{cases}$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

- $\Omega(\sigma_{i=v}\, e, t) \rightsquigarrow \begin{cases} \Omega(e,t) \wedge =(t(i), v) & \text{if } t(i) \neq \mathbf{N} \\ \mathbf{false} & \text{otherwise} \end{cases}$
- $\Omega(\sigma_{i=j}\, e, t) \rightsquigarrow \begin{cases} \Omega(e,t) \wedge =(t(i), t(j)) & \text{if } t(i), t(j) \neq \mathbf{N} \\ \mathbf{false} & \text{otherwise} \end{cases}$
- $\Omega(\pi_{i_1 \cdots i_k} e, t) \rightsquigarrow \exists x_1 \cdots x_n \bigvee_{H \subseteq \{1 \cdots n\} \setminus \{i_1 \cdots i_k\}} \Omega(e, t_H)$
  where x_i are fresh variable symbols and t_H is a sequence of n terms defined as:
  $$t_H(i) \doteq \begin{cases} t(i) & \text{if } i \in \{i_1, \ldots, i_k\} \\ \mathbf{N} & \text{if } i \in H \\ x_i & \text{otherwise} \end{cases}$$
- $\Omega(e_1 \times e_2, t) \rightsquigarrow \Omega(e_1, t') \wedge \Omega(e_2, t'')$
  where n_1, n_2 are the arity of e_1, e_2 respectively,
  where t' is a n_1-ary tuple function s.t. $t'(i) = t(i)$ for $1 \leq i \leq n_1$,
  and t'' is a n_2-ary tuple function s.t. $t''(i) = t(n_1 + i)$ for $1 \leq i \leq n_2$
- $\Omega(e_1 \cup e_2, t) \rightsquigarrow \Omega(e_1, t) \vee \Omega(e_2, t)$
- $\Omega(e_1 - e_2, t) \rightsquigarrow \Omega(e_1, t) \wedge \neg\Omega(e_2, t)$

Example 4. Given some database instance, checking whether the tuple $\langle a, b\rangle$ or the tuple $\langle b, \mathbf{N}\rangle$ are in the answer of the $\mathcal{RA}^{\mathbf{N}}$ query $\sigma_{1=1}\ \sigma_{2=2}\ R$ (discussed in Section 1) corresponds to check the satisfiability over the database instance of the $\mathcal{FOL}^{\varepsilon}$ closed safe-range formula $R(a, b)$ in the former case, or of the formula **false** in the latter case. You can observe that, as expected, also when translated in first-order logic, it turns out that the tuple $\langle b, \mathbf{N}\rangle$ is not in the answer of the query $\sigma_{1=1}\ \sigma_{2=2}\ R$ for any database.

Example 5. Let's consider the "UNIQUE constraint for $R.1$" from Example 1 expressed in $\mathcal{RA}^{\mathbf{N}}$ as $\sigma_{1=3}\ \sigma_{2\neq 4}\ (R \times R) = \emptyset$. The $\mathcal{RA}^{\mathbf{N}}$ expression is translated as the closed safe-range $\mathcal{FOL}^{\varepsilon}$ formula: $\forall x, y, z.\ R(x, y) \wedge R(x, z) \rightarrow y = z$, which corresponds to the way to express a *unique* constraint in first-order logic.

Let's see the translation in three steps from safe-range $\mathcal{FOL}^{\varepsilon}$ formulae to $\mathcal{RA}^{\mathbf{N}}$ expressions. First, the $\mathcal{FOL}^{\varepsilon}$ formula over the signature $\mathcal{R}$ is first translated into a safe-range $\mathcal{FOL}$ formula over the signature $\widetilde{\mathcal{R}}$, as explained in Section 4.1. Then, the safe-range $\mathcal{FOL}$ formula over the signature $\widetilde{\mathcal{R}}$ is translated into a $\mathcal{RA}$ expression over the signature $\widetilde{\mathcal{R}}$, using the classical translation of classical relational algebra into safe-range first-order logic [3]. Finally, the $\mathcal{RA}$ expression over the signature $\widetilde{\mathcal{R}}$ is translated into a $\mathcal{RA}^{\mathbf{N}}$ expression over the signature $\mathcal{R}$, where each basic relation $\widetilde{R}_{\{i_1,\ldots,i_k\}}$ with R of arity n is substituted according to the rule:

$$\widetilde{R}_{\{i_1,\ldots,i_k\}} \rightsquigarrow \pi_{i_1,\ldots,i_k}(\sigma_{\mathsf{isNotNull}(\{i_1,\ldots,i_k\})}\ (\sigma_{\mathsf{isNull}([1\cdots n]\setminus\{i_1,\ldots,i_k\})}\ R))$$

where $\sigma_{\mathsf{isNull}(i_1,\ldots,i_k)}\ e$ is a shorthand for $\sigma_{\mathsf{isNull}(i_1)} \cdots \sigma_{\mathsf{isNull}(i_k)}\ e$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

Example 6. The safe-range closed $\mathcal{FOL}^{\varepsilon}$ formula $\exists x.R(x, \mathbf{N})$ is translated as the zero-ary $\mathcal{RA}^{\mathbf{N}}$ statement $\sigma_{\mathsf{isNotNull}(1)}\ \sigma_{\mathsf{isNull}(2)}\ R \neq \emptyset$.

Example 7. Let's consider what would be the classical way to express the "FOREIGN KEY from $S.2$ to $R.1$" constraint from Example 1 in first-order logic:

$$\forall x, y.\ S(x, y) \rightarrow \exists z.\ R(y, z) \quad \equiv \quad \neg\exists y.\ (\exists x.\ S(x, y)) \wedge \neg(\exists z.\ R(y, z)).$$

The formula is translated as the $\mathcal{RA}^{\mathbf{N}}$ statement: $\pi_2\sigma_{\mathsf{isNotNull}(2)}\ S - \pi_1\sigma_{\mathsf{isNotNull}(1)}\ R = \emptyset$, which is the $\mathcal{RA}^{\mathbf{N}}$ statement we considered in Example 1.

6 Semantic Integrity with Null Values in SQL:1999

In this Section we consider the main integrity constraints involving a specific behaviour for null values as defined in the ANSI/ISO/IEC standard SQL:1999, and we show how these can be naturally captured using $\mathcal{FOL}^{\varepsilon}$. We focus on *unique and primary key constraints* and on *foreign key constraints* as defined in [6]. We will see how the actual definitions of the unique, primary key, and foreign key constraints are a bit more involved than we have seen before: this is because more complex cases than the simple examples above may happen involving null values.

Unique and primary key constraints. As specified in [6], a uniqueness constraint $\mathbf{UNIQUE}(u_1, \ldots, u_n)$ holds for a table R of arity $m > n$ in a database if and only if there are no two rows r_1, r_2 in R such that the values of all their uniqueness columns u_i match and are not null. More formally, in tuple-relational calculus with explicit null values in the domain (fixing the incomplete definition in [6]):

$$\forall r_1, r_2 \in R.\ \left(r_1 \neq r_2 \wedge \bigwedge_{i=1}^{n} r_1.u_i \neq \mathbf{N} \wedge r_2.u_i \neq \mathbf{N}\right) \rightarrow \left(\bigvee_{i=1}^{n} r_1.u_i \neq r_2.u_i\right).$$

In $\mathcal{FOL}^{\varepsilon}$ this constraint can be written as:

$$\forall \overline{x}.\ \left(\bigwedge_{\{i_1 \cdots i_k\} \subseteq \{1 \cdots m\} \setminus \{u_1 \cdots u_n\}} \forall \overline{y}, \overline{z}.\ (\Pi_{u_1,\ldots,u_n,i_1,\ldots,i_k} R(\overline{x}, \overline{y}) \wedge \Pi_{u_1,\ldots,u_n,i_1,\ldots,i_k} R(\overline{x}, \overline{z})) \rightarrow \overline{y} = \overline{z}\right) \wedge$$

$$\forall \overline{x}.\ \left(\bigwedge_{\{i_1 \cdots i_k\} \subseteq \{1 \cdots m\} \setminus \{u_1 \cdots u_n\}} \exists \overline{y}.\ \Pi_{u_1,\ldots,u_n,i_1,\ldots,i_k} R(\overline{x}, \overline{y})\right) \rightarrow \bot$$

where $\Pi_{i_1,\ldots,i_k} R(t_1, \ldots, t_k)$ is a shorthand for $R(t'_1, \ldots, t'_m)$ where $t'_j = t_j$ for $j = i_1 \ldots i_k$ and $t'_j = \mathbf{N}$ for all other j.

A *primary key* constraint is a combination of a *uniqueness* constraint and one or more *not null* constraints. A constraint $\mathbf{PRIMARY\ KEY}(u_1, \ldots, u_n)$ holds for a table R if and only if the following holds, in tuple-relational calculus with explicit null values in the domain (fixing the incomplete definition in [6]):

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

$$\forall r \in R.\ \left(\bigwedge_{i=1}^{n} r.u_i \neq \mathbf{N}\right) \wedge \forall r_1, r_2 \in R.\ r_1 \neq r_2 \rightarrow \left(\bigvee_{i=1}^{n} r_1.u_i \neq r_2.u_i\right).$$

Note that no column shall occur more than once within the same unique/primary key definition; furthermore, each table can have at most one primary key.
In $\mathcal{FOL}^{\varepsilon}$ this constraint can be written as the conjunction of a uniqueness constraint as above and a not null constraint as follows for each key attribute u_i:

$$\neg\left(\bigvee_{\{i_1\cdots i_k\}\subseteq\{1\cdots m\}\setminus\{u_i\}} \exists\overline{y}.\ \Pi_{i_1,\ldots,i_k} R(\overline{y})\right).$$

Foreign key constraints. Foreign key constraints (or referential constraints) express dependencies among (portions of) rows in tables. Given a *referenced* (or *parent*) table and a *referencing* (or *child*) table, a subset $f_i, \ldots, f_k$ of the columns of the referencing table builds the foreign key and refers to the unique/primary key columns $u_j, \ldots, u_l$ of the referenced table. As specified in [6], the *simple match* is the default foreign key constraint implemented by all DBMS vendors. For each row r of the referencing table R (child table), either at least one of the values of the referencing columns $f_1, \ldots, f_n$ is a null value or the value of each referencing columns $f_i, 1 \leq i \leq n$, is equal to the value of the corresponding referenced column u_i for some row s of the referenced table S. More formally, in tuple-relational calculus with null values in the domain [6]:

$$\forall r \in R.\ \left(\bigwedge_{i=1}^{n} r.f_i \neq \mathbf{N}\right) \rightarrow \exists s \in S.\ \left(\bigwedge_{i=1}^{n} r.f_i = s.u_i\right).$$

In $\mathcal{FOL}^{\varepsilon}$ this constraint can be written as:

$$\forall\overline{x}.\ \left(\bigvee_{\{i_1\cdots i_k\}\subseteq\{1\cdots m\}\setminus\{u_1\cdots u_n\}} \exists\overline{y}.\ \Pi_{u_1,\ldots,u_n,i_1,\ldots,i_k} S(\overline{x},\overline{y})\right) \rightarrow \left(\bigvee_{\{i_1\cdots i_k\}\subseteq\{1\cdots m\}\setminus\{u_1\cdots u_n\}} \exists\overline{y}.\ \Pi_{u_1,\ldots,u_n,i_1,\ldots,i_k} R(\overline{x},\overline{y})\right).$$

We observe that, if the database does not contain null values, the SQL:1999 definitions of unique, not null, and foreign key constraints (with simple match) involving null values reduce to the well known classical $\mathcal{FOL}$ definitions of these constraints without null values.

7 Conclusions

Since their inception, SQL null values have been at the centre of long discussions about their real meaning and their formal semantics (see, e.g., [7]). The vast

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

majority of logic based approaches consider nulls as values with an unknown interpretation (i.e., a value exists but it is not known) and they model them as existential variables (e.g. naive tables [1] and the works inspired by [8]). In spite of the fact that these works have their merits and provide a well founded characterisation of incomplete information in databases, they diverge from SQL standard. We have shown that null values – when defined as in SQL with the three-valued logic – should be interpreted as *nonexistent* values, and that such null values do not introduce any incompleteness in the data. In this paper we extend Codd's theorem stating the equivalence of the relational algebra with the relational calculus, to the case in which SQL null values are added to the language.

References

1. Imieliński, T., Lipski, Jr., W.: Incomplete information in relational databases. J. ACM **31**(4) (September 1984) 761–791
2. Date, C.J., Darwen, H.: Database Explorations: Essays on the Third Manifesto and Related Topics. Trafford Publishing (2010)
3. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
4. Codd, E.F.: Extending the database relational model to capture more meaning. ACM Trans. Database Syst. **4**(4) (1979) 397–434
5. Date, C.: An Introduction to Database Systems. 8 edn. Addison-Wesley (2003)
6. Türker, C., Gertz, M.: Semantic integrity support in SQL:1999 and commercial (object-)relational database management systems. The VLDB Journal **10**(4) (2001) 241–269
7. Grant, J.: Null values in SQL. SIGMOD Rec. **37**(3) (September 2008) 23–25
8. Zaniolo, C.: Database relations with null values. Journal of Computer and System Sciences **28**(1) (1984) 142–166

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

An Extension of Datalog for Graph Queries*

Mirjana Mazuran[1], Edoardo Serra[2], and Carlo Zaniolo[3]

[1] Politecnico di Milano DEI – mazuran@elet.polimi.it
[2] University of Calabria DEIS – eserra@deis.unical.it
[3] University of California, Los Angeles – zaniolo@cs.ucla.edu

Abstract. Supporting aggregates in recursive logic rules is a crucial long-standing problem for Datalog. To solve this problem, we propose Datalog^{FS} that supports queries and reasoning on the number of distinct occurrences satisfying given goals, or conjunction of goals, in rules. By using a generalized notion of multiplicity called frequency, we show that graph queries can be easily expressed in Datalog^{FS}. This simple extension preserves all the desirable semantic and computational properties of logic-based languages, while significantly extending their application range to support efficiently page-rank, and social-network queries.

1 Introduction

Due to the emergence of many important application areas we are now experiencing a major resurgence of interest in Datalog for parallel and distributed programming [1] and for expressing and supporting subsets of Description Logic for ontological queries [2]. Other lines of work are exploring execution of recursive queries in the MapReduce framework [3] and in Data Stream Management Systems [4]. The abundance of new applications underscores the need to tackle and solve crucial Datalog problems that remain unsolved and restrict its effectiveness by e.g., disallowing the use of aggregates in recursion. This problem is very challenging since basic aggregates violate the requirement of monotonicity on which the least fixpoint semantics of Datalog is based.

Related Work The notion of *stratification* with respect to negation and aggregates is simple for users to master [5, 6]. Unfortunately, stratification (into a finite number of strata) is too restrictive and cannot support the efficient formulation of many graph optimization algorithms, which typically require the use of extrema and counting in recursion [7].

The importance of optimization and graph applications have motivated much research work seeking to solve these problems. These proposals follow three main approaches: *i*) supporting infinite levels of stratifications using Datalog$_{1S}$ programs [6]; *ii*) attempting to preserve the fixpoint computation via continuous aggregates and non-deterministic *choice* constructs [8, 9], and *iii*) seeking to achieve monotonicity by using partial orders that are more general than set-containment [10]. These past solutions had limited generality and often required

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

sophisticated users and compilers. We next introduce Datalog^{FS} which does not suffer from these problems.

In the next section we introduce Datalog^{FS} via simple examples. In Section 3 we introduce constructs that support facts and predicates having multiple occurrences and, in Section 4, we review important graph applications. In Section 5, we show how to implement Datalog^{FS} efficiently.

2 Datalog^{FS} by Example

Consider, for instance a database of facts as follows:

person(adam). person(marc). person(jerry). person(tom).
son(marc, tom). son(marc, jerry). son(tom, eddy). son(tom, adam). son(tom, john).

The following rule defines fathers with at least two sons:

$$\texttt{twosons(X)} \leftarrow \texttt{person(X)}, \texttt{son(X, Y1)}, \texttt{son(X, Y2)}, \texttt{Y2} \neq \texttt{Y1}.$$

Datalog^{FS} allows the following equivalent expression for our twosons rule:

$$\texttt{twosons(X)} \leftarrow \texttt{person(X)},\ \texttt{2:[son(X, Y)]}.$$

The goal, I:[b-expression], is called a frequency support goal (or *FS-Goal*), and "I" is a positive integer, called *Running-FS clause*. The expression in the bracket is called *b-expression*, and can either consists of a single positive predicate or a conjunction of positive predicates [11]. The convenience of FS-goals is clear if we want to find people with many sons:

$$\texttt{sixsons(X)} \leftarrow \texttt{person(X)},\ \texttt{6:[son(X, Y)]}.$$

will retrieve all persons who have at least six sons. An equivalent rule can be expressed using the $\neq$ operator. Indeed we can start as follows:

$$\texttt{sixsons(X)} \leftarrow \texttt{person(X)}, \texttt{son(X, Y1)}, \texttt{5:[son(X, Y2)}, \texttt{Y2} \neq \texttt{Y1]}.$$

and proceed inductively, and obtain a rule containing six goals $\texttt{son(X, Y}_\texttt{j}\texttt{)}$, where $\texttt{j} = 1, \ldots, 6$ and 6×5 goals saying, that every Y be different from every other Y. If we are interested in links between web pages, which could easily be thousands, it becomes clear that the approach based on $\neq$ is totally impractical, and without FS-goals we would need a COUNT aggregate. Yet, aggregates bring in the curse of non-monotonicity and recursion becomes a problem. At the semantic level, our Datalog^{FS} rules can instead be viewed as standard Horn clauses whereby the standard monotonicity-based semantics of negation-free Datalog is preserved.

We now clarify the scope of variables in Datalog^{FS}. Predicate friend(X, Y) denotes that person X views person Y as a friend (no assumption of symmetry):

Example 1. Pairs of friends (F1, F2) where F1 and F2 have at least 3 friends:

$$\texttt{popularpair(X, Y)} \leftarrow \texttt{friend(X, Y)}, \texttt{3:[friend(X, V1)]}, \texttt{3:[friend(Y, V2)]}.$$

Example 2. Pairs of friends (F1, F2) who have at least three friends in common

$$\texttt{sharethree(X, Y)} \leftarrow \texttt{friend(X, Y)}, \texttt{3:[friend(X, V)}, \texttt{friend(Y, V)]}.$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

There are two kinds of variables in rules with FS-goals. The first are those, such as `X` and `Y` in Example 2, that appear in the head of the rule or in some goal outside the b-expression. These will be called *global* variables. They are basically the universally qualified variables of the standard Horn Clauses, and have the whole rule as scope. Variables `X` and `Y` in Example 1 are global for that rule.

Other variables only appear in b-expressions and their scope is *local* to the b-expression, where they appear (e.g. `V1` and `V2` in Example 1, and `V` in Example 2). Thus, $K[\ldots]$ can be viewed as an existential declaration of local variables under the following constraint: there exist at least K assignments of the local variables that satisfy the b-expression. Example 2 states that there exist at least 3 distinct `V` occurrences each denoting a person who is a friend to both `X` and `Y`. The scope of existential variables is local to the b-expression: in Example 1 replacing `V1` and `V2` with `V` would not change the meaning of our rule. Let us now express that an assistant professor to be advanced to associate professor should have an H-index of at least 13:

Example 3. Our candidate must have authored at least 13 papers each of which has been referenced at least 13 times. The database table `author(Author, Pno)` lists all papers (co-)authored by a person, while the atom `refer(PnFrom, PnTo)` denotes that paper `PnFrom` contains a reference to paper `PnTo`.

```
atleast13(PnTo) ← 13:[refer(PnFrom, PnTo)].
hindex13(Author) ← 13:[author(Author, Pno), atleast13(Pno)].
```

These simple examples could also be expressed using the count aggregate. Yet, count and other aggregates are non-monotonic with respect to the partial ordering defined by set containment, and cannot be used in recursive rules. Indeed, the meaning and efficient implementation of Datalog programs with recursive rules are based on their least fixpoint semantics[4], which is only guaranteed to exist when the program rules define mononotonic mappings. Now, continuous count is obviously monotonic with respect to set-containment. This is formally proven in [11] by rewriting the running FS-construct with equivalent, although inefficient, Horn clauses (that use list and thus we will avoid in the actual implementation, as shown in Section 5).

Final-FS construct and Stratification. The semantics of Datalog^{FS} [11] allows to use variables rather than constants in the specification of FS goals. This is useful, for instance, to find the actual number of sons a person has:

Example 4. How many sons does a person have?

```
csons(PName, N) ← person(PName), N:[son(PName, Sname)], ¬morethan(PName, N).
morethan(PName, N) ← N1:[son(PName, _)], N1 > N.
```

Thus `csons` must belong to a stratum that is strictly higher than `son`, whereas with respect to `person` it could be in the same stratum or in the one above it. The need to find the maximum value satisfying a running-FS clause is so common that we provide a construct called Final-FS and denoted by the operator `=!`.

[4] Naturally, by "least fixpoint" of a program, we mean "least fixpoint of its immediate consequence operator" [12].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Example 5. How many sons does a person have?

$$\texttt{csons(Name, N)} \leftarrow \texttt{person(Name)}, \texttt{N} = ![\texttt{son(Name, \_)}].$$

The formal semantics of the *Final-FS* construct is defined as the rewriting of Example 5 into Example 4, which makes use of negation, whereby we will require that our Datalog^{FS} programs be stratified w.r.t. Final-FS goals.

Recursive Datalog^{FS}. Consider the following example:
Example 6. Some people will come to the party for sure. Others will also come once they learn that three or more of their friends will come.

$$\texttt{willcome(X)} \leftarrow \texttt{sure(X)}.$$
$$\texttt{willcome(Y)} \leftarrow \texttt{3}:[\texttt{friend(Y, X)}, \texttt{willcome(X)}].$$

One person might be more timid than another, and different people could require a different number of friends before they also join the party. Thus, if `requires(Person, PNumber)` denotes the number of friends required by a person, where `PNumber` must be a positive integer (whereas `sure` denotes people who will come even if none of their friends will), we have the following program:

Example 7. A person will join the party if a sufficient number of friends join.

$$\texttt{join(X)} \leftarrow \texttt{sure(X)}.$$
$$\texttt{join(Y)} \leftarrow \texttt{requires(Y, K)}, \texttt{K}:[\texttt{friend(Y, X)}, \texttt{join(X)}].$$

3 Multi-Occuring Predicates

As discussed in [10], there are numerous examples where it is desirable that certain predicates are counted as providing a support level greater than one. For instance, we might use the following representation to denote that the paper with DBLP identifier "MousaviZ11" is currently cited in six papers: `ref("MousaviZ11"):6`. Thus, `Pno` = "`MousaviZ11`" contributes with count six to the b-expression of the rule:

Example 8. Total reference count for an author.

$$\texttt{tref(Authr)}:\texttt{N} \leftarrow \texttt{N}:[\texttt{author(Authr, Pno)}, \texttt{ref(Pno)}].$$

The clauses `":6" and ":N"` used in the above fact and rule head will be called *FS-Assert* clauses. The semantics of programs P with frequency assert clauses is defined by *expanding* it into its $\bar{P}$ equivalent, which is obtained as follows: Each rule in P with head $\texttt{q(X1}, \ldots, \texttt{Xn)}:\texttt{K} \leftarrow \texttt{Body}$ is replaced by

$$\bar{\texttt{q}}(\texttt{X1}, \ldots, \texttt{Xn}, \texttt{J}) \leftarrow \texttt{lessthan(J, K)}, \texttt{Body}.$$

where `lessthan(J, K)` is a recursive predicate that generates all positive integers up to `K`, included. Thus, we have that, as a result of this expansion, `ref("MousaviZ11"):6` contributes with six to the reference count of each author of that paper. A property of frequency statements is that, when multiple statements hold for the same fact *only the largest value* is significant, the others are subsumed and can be ignored.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Bill-of-materials (BOM) applications represent a well-known example of the need for recursive queries. Our database might contain records assbl(Part, Subpart, Qty) which, for each part number, gives the immediate subparts used in its assembly and their quantity (e.g. a bicycle has 1 frame and 2 wheels as immediate subparts). At the bottom of BOM DAG, we find the basic parts that are purchased from external suppliers and described by basic(Pno, Days) denoting the days needed to obtain that basic part. Several interesting BOM applications are naturally expressed by combining aggregates and recursion, as follows:

Example 9. How many basic parts does an assembled part contain?

$$\texttt{cassb(Part, Sub):Qty} \leftarrow \texttt{assbl(Part, Sub, Qty)}.$$

$$\texttt{cbasic(Pno):1} \leftarrow \texttt{basic(Pno, \_)}.$$
$$\texttt{cbasic(Part):K} \leftarrow \texttt{K:[cassb(Part, Sub), cbasic(Sub)]}.$$
$$\texttt{cntbasic(Prt, C)} \leftarrow \texttt{C =![cbasic(Prt)]}.$$

The count of basic parts is not retrieved using the goal N : [cbasic(frame)] since this will return all positive integers up to the max value. Goal N =![cbasic(frame)] is used instead as this returns the exact count of the basic subparts.

Example 10. How many days until delivery?

$$\texttt{delivery(Pno):Days} \leftarrow \texttt{basic(Pno, Days)}.$$
$$\texttt{delivery(Part):Days} \leftarrow \texttt{assb(Part, Sub, \_), Days:[delivery(Sub)]}.$$
$$\texttt{actualDays(Part, CDays)} \leftarrow \texttt{CDays =![delivery(Part)]}.$$

For each assembled part, we find each basic subpart along with the number of days this takes to arrive. Observe that the argument Pno is projected out, and only the number of days associated with it is retained, whereby the maximum of the number of days required by any basic part is derived.

4 Advanced Graph Applications

We now show some examples that use Datalog^{FS} with positive rational numbers. As explained in more details in [11], we can assume that we use rational numbers with the same large denominator, and thus easily derive equivalent Datalog^{FS} rules for their numerators.

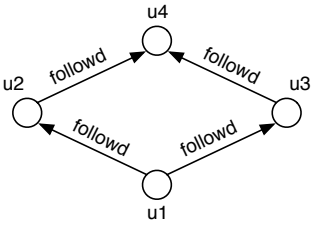
Diffusion Models. The Jackson-Yariv Diffusion Model (JYDM) [13] provides a powerful abstraction on how social structures influence the spread of a behavior and trends in Social Networks. We use the JDYM to understand how a tweet will spread in the Twitter network. Let followd(X, Y) indicate that user X is followed by user Y and coeff(X, C) means that C is the coefficient of how susceptible to change node X is. Predicate b(X) denotes, if true, that node X will retweet. We assume that an agent, source(X), first posts the tweet and starts its diffusion.

Example 11. Modeling a retweet in Datalog^{FS}.

$$\texttt{b(X)} \leftarrow \texttt{source(X)}.$$
$$\texttt{b(X)} \leftarrow \texttt{coeff(X, C), K}\geq\texttt{1/C, K:[follwd(Y, X), b(Y)]}.$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

The last rule finds each node X for which the condition $\mathtt{K} \times \mathtt{C} \geq 1$ holds by testing the equivalent condition $\mathtt{K} \geq 1/\mathtt{C}$. When this is satisfied $\mathtt{b}(\mathtt{X})$ is set to true. The answer to the query $?\mathtt{b}(\mathtt{Y})$ will thus list all the users that propagated the tweet that originated from the node specified by `source`.




```
followd(u1,u2).   coeff(u2,1.08).   source(u1).
followd(u1,u3).   coeff(u3,0.60).
followd(u2,u4).   coeff(u4,1.15).
followd(u3,u4).
```

Fig. 1. Retweet modeling in Datalog^{FS}.

Then, applying the program in Example 11 to the Twitter network in Figure 1 the following atoms are derived: $\mathtt{b}(\mathtt{u}_1), \mathtt{b}(\mathtt{u}_2), \mathtt{b}(\mathtt{u}_4)$.

Markov Chains and Page Rank. A Markov chain is represented by the transition matrix W of $s \times s$ components where w_{ij} is the probability to go from state i to state j in one step. A Markov chain is *irreducible* if for each pair of states i, j, the probabilities to go from i to j and from j to i in one or more steps is greater than zero.

Computing stabilized probabilities of a Markov chain has many real-world applications, such as estimating the distribution of population in a region, and determining the Page Rank of web nodes. Let P be a vector of stabilized probabilities of cardinality s, the equilibrium condition in terms of matrices is: $P = W \cdot P$.

Computing this fixpoint is far from trivial and irreducible chains can be modeled quite naturally in Datalog^{FS}. If $\mathtt{p\_state}(\mathtt{X}) : \mathtt{K}$ denotes that $\mathtt{K}$ is the probability of node $\mathtt{X}, 1 \leq \mathtt{X} \leq \mathtt{s}$, and $\mathtt{w\_matrix}(\mathtt{Y}, \mathtt{X}) : \mathtt{W}$ denotes that the arc from $\mathtt{Y}$ to $\mathtt{X}$ has weight $\mathtt{W}$, then we compute the fixpoint as follows:

$$
\begin{aligned}
&\mathtt{p\_state}(\mathtt{X}) : \mathtt{K} \leftarrow \quad \mathtt{K} : [\mathtt{p\_state}(\mathtt{Y}), \mathtt{w\_matrix}(\mathtt{Y}, \mathtt{X})].\\
&\mathtt{w\_matrix}(1, 1) : \mathtt{w}_{11}.\\
&\mathtt{w\_matrix}(1, 2) : \mathtt{w}_{12}.\\
&\vdots\\
&\mathtt{w\_matrix}(\mathtt{s}, \mathtt{s}) : \mathtt{w}_{\mathtt{ss}}.
\end{aligned}
$$

It is important to notice that each fixpoint of such program is an equilibrium $P = W \cdot P$ of the Markov Chain represented by matrix W. In order to find a non trivial fixpoint ($\neq 0$) for program P, we add baseline facts i.e. a set of facts $\mathtt{p\_state}(1) : 0.1.\ \mathtt{p\_state}(2) : 0.1.\ \ldots\ \mathtt{p\_state}(\mathtt{s}) : 0.1.$, that guarantee that the least fixpoint contains facts with predicate `p_state`. Such program is called *Pbl* and is a Datalog^{FS} program for which we can compute the least fixpoint efficiently. Moreover, every fixpoint of *Pbl* is also a fixpoint for P. Indeed, for any interpretation I that contains all the baseline facts, the application of either operators produce the same result: i.e., $T_P(I) = T_{Pbl}(I)$. Therefore any fixpoint of T_P that contains all the baseline facts is also a fixpoint for T_{Pbl} and vice-versa.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

But since, by its very definition, the least model of *Pbl* contains all the baseline facts, we have that every fixpoint for T_{Pbl} is also a fixpoint for T_P. The opposite of course is not true since the null fixpoint of T_P, and possibly others, are not fixpoint for T_{Pbl}. However, if T_P has a fixpoint that is positive at all nodes, then by multiplying the frequency at all nodes by a large enough finite constant, we obtain a fixpoint for T_P that contains all the baseline facts of T_{Pbl}. Since, for each irreducible Markov chain there exists a not trivial fixpoint, also T_P has one that is not null at every node, then there exists a finite fixpoint for T_{Pbl}. Therefore, the least fixpoint for T_{Pbl} is finite. That is:

Theorem 1.

- *The least fixpoint of the baseline Datalog^{FS} program that models an irreducible Markov chain is finite.*
- *Every non-null solution of an irreducible Markov chain can be obtained by scaling the least fixpoint solution of its baseline Datalog^{FS} model.*

In summary, while there has been a significant amount of previous work on Markov chains, the use of Datalog^{FS} has provides us with a model and a simple computation algorithm which is valid for all irreducible Markov chains, including periodic ones.

5 Efficient Implementation

The greater expressive power of Datalog^{FS} combines with its amenability to efficient implementation via the following three optimization steps: (i) differential fixpoint, (ii) Magic Sets, and (iii) Max-optimization. Since (ii) is basically the same as that in Datalog [11], we will discuss here (i) and (iii).

In Datalog^{FS} the differential fixpoint step is applied to recursive rules after they are transformed to ensure that every goal in the b-expression also appears outside the bracket. To satisfy this requirement, the second rule in Example 7 is transformed by repeating outside the bracket the two clauses inside the bracket, producing the following rule:

$$\texttt{join(Y)} \leftarrow \texttt{requires(Y,K)}, \texttt{friend(Y,X1)}, \texttt{join(X1)}, \texttt{K:[friend(Y,X), join(X)]}.$$

Since we have renamed the local variables (`X` for the case at hand), and since $\texttt{K} \geq 1$, this transformation does not change the meaning of the rule. However it greatly simplifies its symbolic differentiation since the bracketed expression can now be treated as a constant. Thus the rule becomes linear and its δ version is:

$$\delta\texttt{join(Y)} \leftarrow \texttt{requires(Y,K)}, \texttt{friend(Y,X1)}, \delta\texttt{join(X1)}, \texttt{K:[friend(Y,X), join(X)]}.$$

The Max-optimization transforms the delta rules so obtained by replacing the running FS-construct with the final FS-construct. For instance, we start by rewriting the delta rule above into the following one that preserves its operational semantics:

$$\delta\texttt{join(Y)} \leftarrow \texttt{requires(Y,K1)}, \texttt{friend(Y,X1)}, \delta\texttt{join(X1)}, \texttt{K:[friend(Y,X), join(X)]}, \texttt{K} \geq \texttt{K1}.$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Now, we can replace the running-FS `K`:[...] by the final-FS `K` =![...] and still preserve the operational semantics of the rule, due to the fact that the rule only uses `K` in monotonic functions and predicates (e.g., predicates that if they are true for `K` they are also true for every value larger than `K`). This optimization would not be possible if the body uses some non-monotonic predicate, e.g., a goal that checks that `K` is even. A similar situation occurs in Examples 9, where instead of the running-FS construct in the recursive δ rules we can use the final-FS construct. Indeed the values produced by the former satisfy the external goal `C` =![`cbasic`(`part`)] iff the values produced by the latter do. Again this equivalence is due to the monotoniciy of the arithmetic and boolean predicates used in the rule. Such monotonicity holds in all examples given in this paper and the many examples of practical interest discussed in [11].

6 Conclusion

In this paper, we studied the important problem of allowing aggregates in recursive Datalog rules, and proposed a solution of surprising simplicity for this long-standing challenge. Our Datalog^{FS} approach is based on using continuous aggregate-like functions that allow us to query and reason on the frequency with which predicates and conjunctions of predicates occur.

References

1. J. M. Hellerstein. Datalog redux: experience and conjecture. In *PODS*, pages 1–2, 2010.
2. G. Gottlob, G. Orsi, and A. Pieris. Ontological queries: Rewriting and optimization. In *ICDE*, pages 2–13, 2011.
3. F. N. Afrati, V. R. Borkar, M. J. Carey, N. Polyzotis, and J. D. Ullman. Map-reduce extensions and recursive queries. In *EDBT*, pages 1–8, 2011.
4. C. Zaniolo. The logic of query languages for data streams. In *Logic and Databases 2011. EDBT 2011 Workshops*, pages 1–2, 2011.
5. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
6. C. Zaniolo, S. Ceri, C. Faloutsos, R. T. Snodgrass, V. S. Subrahmanian, and R. Zicari. *Advanced Database Systems.* Morgan Kaufmann, 1997.
7. I. S. Mumick and O. Shmueli. How expressive is stratified aggregation? *Annals of Mathematics and Artificial Intelligence*, 15:407–435, 1995.
8. S. Greco and C. Zaniolo. Greedy algorithms in datalog. *TPLP*, 1(4):381–407, 2001.
9. F. Arni, K. Ong, S. Tsur, H. Wang, and C. Zaniolo. The deductive database system ldl++. *TPLP*, 3(1):61–94, 2003.
10. I. S. Mumick, H. Pirahesh, and R. Ramakrishnan. The magic of duplicates and aggregates. In *VLDB*, pages 264–277, 1990.
11. M. Mazuran, E. Serra, and C Zaniolo. Graph languages in Datalog^{FS}: from abstract semantics to efficient implementation. Technical report, UCLA, 2011.
12. J. W. Lloyd. *Foundations of Logic Programming, 2nd Edition.* Springer, 1987.
13. M. O. Jackson and L. Yariv. Diffusion on social networks. *Economie Publique*, 2005.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Stratification-based Criteria for Checking Chase Termination*

Sergio Greco, Francesca Spezzano, and Irina Trubitsyna

DEIS, Università della Calabria, 87036 Rende, Italy
{greco,fspezzano,irina}@deis.unical.it

Abstract. Several database areas such as data exchange and integration share the problem of fixing database instance violations with respect to a set of constraints. The chase algorithm solves such violations by inserting tuples and setting the value of nulls. Unfortunately, the chase algorithm may not terminate and the problem of deciding whether the chase process terminates is undecidable. Recently there has been an increasing interest in the identification of sufficient structural properties of constraints which guarantee that the chase algorithm terminates.
In this paper we present more general criteria for chase termination. We first present extensions of the well-known stratification condition and, then, introduce a new criterion, called local stratification (LS), which generalizes both super-weak acyclicity and stratification-based criteria (including the class of constraints which are inductively restricted).

1 Introduction

Several database areas such as data exchange and integration share the problem of fixing database instance violations with respect to a set of constraints [1–6]. The chase algorithm solves such violations by inserting tuples and setting the value of nulls. Unfortunately, the chase algorithm may not terminate and the problem of deciding whether the chase process terminates is undecidable. Recently there has been an increasing interest in the identification of sufficient structural properties of constraints which guarantee that the chase algorithm terminates. Most of these criteria extend weak acyclicity (WA) [7] by analyzing nulls propagation (e.g. the safety criterion (SC) [8]) and constraints firing (e.g. c-stratification (CStr) and inductive restriction (IR) [9], super-weak acyclicity (SwA) [10]).

The idea underlying c-stratification, also used in the IR and SwA criteria, is to consider, in the propagation of nulls, how constraints may fire each other. However, there are simple cases where current criteria are not able to understand that all chase sequences are finite.

Example 1. Consider the following set of constraints Σ_1 consisting of the TGD

$$\forall x \forall y \; E(x,y) \wedge E(y,x) \rightarrow \exists z \; E(y,z) \wedge E(z,x)$$

We can distinguish two cases. If we suppose to have a database instance $D_1 = \{E(a,b), E(b,a)\}$ we have that the TGD is not satisfied, but the constraint will be applied only once by the chase as the resulting database $D_1' = \{E(a,b), E(b,a),$

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

$E(b, \eta_1), E(\eta_1, a)\}$ is consistent. Otherwise, if we suppose to have a database instance $D_2 = \{E(a, a)\}$, it is already consistent, and no chase step will be applied. It is easy to see that the chase is always terminating for all database instances, but none of the existing criteria guaranteeing the termination of all chase sequences is able to recognize it as terminating. □

In order to cope with this problem, we first propose a new extension of c-stratification, called *WA-stratification* (*WA-Str*) and then introduce a new criterion, called *local stratification* (*LS*), which generalizes both super-weak acyclicity and inductive restriction). Moreover, both *WA-Str* and *LS* guarantee the termination of all chase sequences, for all database instances, in polynomial time.

2 Preliminaries

We introduce the following disjunct sets of symbols: (i) an infinite set $Consts$ of *constants*, (ii) an infinite set $Nulls$ of *labeled nulls* and (iii) an infinite set $Vars$ of variables. A *relational schema* $\mathbf{R}$ is a set of relational predicates R, each with its associated arity $ar(R)$. An *instance* of a relational predicate R of arity n is a set of ground atoms in the form $R(c_1, \ldots, c_n)$, where $c_i \in Consts \cup Nulls$. Such (ground) atoms are also called tuples or facts. We denote by D a database instance constructed on $Consts$ and by J, K the database instances constructed on $Consts \cup Nulls$. Given an instance K, $Nulls(K)$ (resp. $Consts(K)$) denotes the set of labeled nulls (resp. constants) occurring in K. An *atomic formula* (or *atom*) is of the form $R(t_1, \ldots, t_n)$ where R is a relational predicate, $t_1, \ldots, t_n$ are terms belonging to the domain $Consts \cup Vars$ and $n = ar(R)$.

Given a relational schema $\mathbf{R}$, a *tuple generating dependency* (TGD) over $\mathbf{R}$ is a formula of the form $\forall \mathbf{x} \forall \mathbf{z}\ \phi(\mathbf{x}, \mathbf{z}) \rightarrow \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})$, where $\phi(\mathbf{x}, \mathbf{z})$ and $\psi(\mathbf{x}, \mathbf{y})$ are conjunctions of atomic formulas over $\mathbf{R}$; $\phi(\mathbf{x}, \mathbf{z})$ is called the *body* of r, denoted as $Body(r)$, while $\psi(\mathbf{x}, \mathbf{y})$ is called the *head* of r, denoted as $Head(r)$. An *equality generating dependency* (EGD) over $\mathbf{R}$ is a formula of the form $\forall \mathbf{x}\ \phi(\mathbf{x}) \rightarrow x_1 = x_2$, where x_1 and x_2 are among the variables in $\mathbf{x}$.

In the following we will often omit the universal quantification, since we assume that variables appearing in the body are universally quantified and variables appearing only in the head are existentially quantified. In some cases we also assume that the head and body conjunctions are sets of atoms.

Definition 1 (Homomorphism [7]). Let K_1 and K_2 be two instances over $\mathbf{R}$ with values in $Consts \cup Nulls$. A *homomorphism* $h : K_1 \rightarrow K_2$ is a mapping from $Consts(K_1) \cup Nulls(K_1)$ to $Consts(K_2) \cup Nulls(K_2)$ such that: (1) $h(c) = c$, for every $c \in Consts(K_1)$, and (2) for every fact $R_i(t)$ of K_1, we have that $R_i(h(t))$ is a fact of K_2 (where, if $t = (a_1, ..., a_s)$, then $h(t) = (h(a_1), ..., h(a_s))$). □

Similar to homomorphisms between instances, a homomorphism h from a conjunctive formula $\phi(\mathbf{x})$ to an instance J is a mapping from the variables $\mathbf{x}$ to $Consts(J) \cup Nulls(J)$ such that for every atom $R(x_1, \ldots, x_n)$ of $\phi(\mathbf{x})$ the fact $R(h(x_1), \ldots, h(x_n))$ is in J.

For any database instance D and set of constraints Σ over a database schema $\mathbf{R}$, a *solution* for (D, Σ) is an instance J such that $D \subseteq J$ and $J \models \Sigma$ (i.e. J satisfies all constraints in Σ). A *universal solution* J is a solution such that for every solution J' there exists a homomorphism $h : J \rightarrow J'$. The set of universal solutions for (D, Σ) will be denoted by $USol(D, \Sigma)$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Definition 2 (Chase step [7]). Let K be a database instance.

(1) Let r be a TGD $\phi(\mathbf{x},\mathbf{z}) \rightarrow \exists \mathbf{y}\, \psi(\mathbf{x},\mathbf{y})$. Let h be a homomorphism from $\phi(\mathbf{x},\mathbf{z})$ to K such that there is no extension of h to a homomorphism h' from $\phi(\mathbf{x},\mathbf{z}) \wedge \psi(\mathbf{x},\mathbf{y})$ to K. We say that r can be *applied* to K with homomorphism h. Let K' be the union of K with the set of facts obtained by: (a) extending h to h' such that each variable in $\mathbf{y}$ is assigned a fresh labeled null, followed by (b) taking the image of the atoms of ψ under h'. We say that the result of applying r to K with h is K', and write $K \overset{r,h}{\rightarrow} K'$.

(2) Let r be an EGD $\phi(\mathbf{x}) \rightarrow x_1 = x_2$. Let h be a homomorphism from $\phi(\mathbf{x})$ to K such that $h(x_1) \neq h(x_2)$. We say that r can be *applied* to K with homomorphism h. More specifically, we distinguish two cases. (a) If both $h(x_1)$ and $h(x_2)$ are in *Consts* the result of applying r to K with h is "failure", and $K \overset{r,h}{\rightarrow} \perp$. (b) Otherwise, let K' be K where we identify $h(x_1)$ and $h(x_2)$ as follows: if one is a constant, then the labeled null is replaced everywhere by the constant; if both are labeled nulls, then one is replaced everywhere by the other. We say that the result of applying r to K with h is K', and write $K \overset{r,h}{\rightarrow} K'$. □

Definition 3 (Chase [7]). Let Σ be a set of TGDs and EGDs, and let K be an instance.

(1) A *chase sequence of K with Σ* is a sequence (finite or infinite) of chase steps $K_i \overset{r,h_i}{\rightarrow} K_{i+1}$, with $i = 0, 1, ...$, $K_0 = K$ and r a dependency in Σ.

(2) A *finite chase of K with Σ* is a finite chase sequence $K_i \overset{r,h_i}{\rightarrow} K_{i+1}$, $0 \leq i < m$, with the requirement that either (a) $K_m = \perp$ or (b) there is no dependency r of Σ and there is no homomorphism h_m such that r can be applied to K_m with h_m. We say that K_m is the result of the finite chase. We refer to case (a) as the case of a *failing finite chase* and we refer to case (b) as the case of a *successful finite chase.* □

In [7] it has been shown that, for any instance D and set of constraints Σ: (i) if J is the result of some successful finite chase of $\langle D, \Sigma \rangle$, then J is a universal solution; (ii) if some failing finite chase of $\langle D, \Sigma \rangle$ exists, then there is no solution.

Chase Termination Criteria. We now present a brief overview on the well-known chase termination conditions that guarantee for every database D the termination of all chase sequences in PTIME in the size of D. Given a criterion C, the class of constraints satisfying C will be denoted by $\mathcal{C}$.

Weak acyclicity. Let Σ be a set of TGDs over a database schema $\mathbf{R}$, then $pos(\Sigma)$ denotes the set of positions R_i such that R denotes a relational predicate of $\mathbf{R}$ and there is an R-atom appearing in Σ. Weak acyclicity (*WA*) is based on the construction of a directed graph $dep(\Sigma) = (pos(\Sigma), E)$, called the *dependency graph*, where E is defined as follows. For every TGD $\phi(\mathbf{x},\mathbf{z}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x},\mathbf{y})$ in Σ, then: i) for every x in $\mathbf{x}$ occurring in position R_i in ϕ and in position S_j in ψ, add an edge $R_i \rightarrow S_j$ (if it does not already exist); ii) for every x in $\mathbf{x}$, appearing in position R_i in ϕ and for every y in $\mathbf{y}$ appearing in position T_k in ψ, add a special edge $R_i \overset{*}{\rightarrow} T_k$ (if it does not already exist). Σ is *weakly acyclic* if $dep(\Sigma)$ has no cycle going through a special edge.

Safety. The safety condition (*SC*) [9] is based on the notion of affected positions. An *affected position* denotes a position in which null values may appear,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

that is it can also take values from $Nulls$. A position R_i is said to be affected if there is a constraint $r : \phi(\mathbf{x}, \mathbf{z}) \rightarrow \exists \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ in Σ and either i) there is a variable y in $\mathbf{y}$ appearing in position R_i in ψ, or ii) there is a variable x in $\mathbf{x}$ appearing both in position R_i in ψ and only in affected positions in the body of r. The set of affected positions of Σ is denoted by $aff(\Sigma)$.

Given a set of TGDs Σ, the *propagation graph* of Σ, denoted as $prop(\Sigma) = (aff(\Sigma), E')$, is a subset of $dep(\Sigma) = (pos(\Sigma), E)$ such that E' contains the edges in E whose positions are affected (since $aff(\Sigma) \subseteq pos(\Sigma)$). Moreover, Σ is said to be safe if $prop(\Sigma)$ does not contain cycles with special edges.

C-Stratification. The idea behind *c-stratification* ($CStr$) [11, 9] is to decompose the set of constraints into independent subsets, where each subset consists of constraints that may fire each other, and check each component separately for weak acyclicity. Given a set of constraints Σ and two constraints $r_1, r_2 \in \Sigma$, we say that $r_1 \prec_c r_2$ iff there exists a relational database instance K_1 and two homomorphisms h_1 and h_2 such that i) $K_1 \xrightarrow{*,r_1,h_1} K_2$, ii) $K_2 \not\models h_2(r_2)$ and iii) $K_1 \models h_2(r_2)$, where the oblivious chase step $K_1 \xrightarrow{*,r_1,h_1} K_2$ states that there is a homomorphism h_1' extending h_1 which associates every existentially variable y in $h_1(r_1)$ to a fresh labeled null. Intuitively, $r_1 \prec_c r_2$ means that firing r_1 can cause the firing of r_2. We say that Σ is c-stratified iff the constraints in every cycle of the *c-chase graph* $G(\Sigma) = (\Sigma, \{(r_1, r_2) | r_1 \prec_c r_2\})$ are weakly acyclic.

Inductive Restriction. The inductive restriction criterion (IR) extends both $CStr$ and SC by partitioning constraints in a more refined way. In particular, it first computes the graph $(G'(\Sigma)) \subseteq G(\Sigma)$ and partition Σ into $\Sigma_1, ..., \Sigma_n$, where each Σ_i is a set of dependencies defining a strongly connected components in $G'(\Sigma)$, next, if $n = 1$ the safety criterion is applied to Σ, otherwise the IR criterion is applied inductively to each Σ_i.

Super-weak acyclicity. The super-weak acyclicity (SwA) [10] builds a *trigger graph* $\Upsilon(\Sigma) = (\Sigma, E)$ where edges define relations among constraints. An edge $r_i \rightsquigarrow r_j$ means that a null value introduced by a constraint r_i is propagated (directly or indirectly) into the body of r_j.

Let Σ be a set of TGDs and let $sk(\Sigma)$ be the logic program obtained by skolemizing Σ, i.e. by replacing each existentially quantified variable y appearing in the head of a TGD r by the skolem function $f_y^r(\mathbf{x})$, where $\mathbf{x}$ is the set of variables appearing both in the body and in the head of r. A *place* is a pair (a, i) where a is an atom of $sk(\Sigma)$ and $0 \leq i \leq ar(a)$. Given a TGD r and an existential variable y in the head of r, $Out(r, y)$ denotes the set of places (called *output places*) in the head of $sk(r)$ where a term of the form $f_y^r(\mathbf{x})$ occurs. Let r be a TGD r and let x be a universal variable of r, $In(r, x)$ denotes the set of places (called *input places*) in the body of r where x occurs.

Given a set of variables V, a substitution θ of V is a function mapping each $v \in V$ to a finite term $\theta(v)$ built upon constants and function symbols. Two places (a, i) and (a', i) are unifiable and we write $(a, i) \sim (a', i)$ iff there exist two substitutions θ and θ' of (respectively) the variables a and a' such that $a[\theta] = a'[\theta']$. Given two sets of places Q and Q' we write $Q \sqsubseteq Q'$ iff for all $q \in Q$ there exists some $q' \in Q'$ such that $q \sim q'$.

For any set Q of places, $Move(\Sigma, Q)$ denotes the smallest set of places Q' such that $Q \subseteq Q'$, and for every constraint $r = B_r \rightarrow H_r$ in $sk(\Sigma)$ and every

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

variable x, if $\Pi_x(B_r) \sqsubseteq Q'$ then $\Pi_x(H_r) \subseteq Q'$, where $\Pi_x(B_r)$ and $\Pi_x(H_r)$ denote the sets of places in B_r and H_r where x occurs.
Given a set Σ of TGDs and two TGDs $r_1, r_2 \in \Sigma$, we say that r_1 triggers r_2 in Σ and write $r_1 \rightsquigarrow r_2$ iff there exists an existential variable y in the head of r_1, and a universal variable x_2 occurring both in the body and head of r_2 such that $In(r_2, x) \sqsubseteq Move(\Sigma, Out(r_1, y))$. A set of constraints Σ is super-weakly acyclic iff the trigger graph $\Upsilon(\Sigma) = (\Sigma, \{(r_1, r_2) | r_1 \rightsquigarrow r_2\})$ is acyclic. W.r.t. other criteria, SwA also takes into account that a variable may occur more than once in the same atom. $\mathcal{SwA}$ extends $\mathcal{SC}$, but is not comparable with $\mathcal{CSt}r$.

3 WA-Stratification

We start by introducing some improvements for the c-stratification criterion. First of all, observe that c-stratification does not specify what kind of cycles are checked (i.e. simple or general) [12]. Checking simple cycles is not correct as it may not consider all possible chase sequences, but checking general cycles, means that for each strongly connected component there is one cycle including all nodes in the component which subsumes all other cycles on the same component (in terms of constraints to be considered). Thus, a first observation on (c-)stratification (in terms of correctness, if simple cycles are considered, or in terms of efficiency, if all cycles are considered) is that it refers to cycles instead of strongly connected components. A further observation is that it uses oblivious chase for checking termination of standard chase and its applicability is limited.

Definition 4 (WA-Stratification). Given a set of dependencies Σ and $r_1, r_2 \in \Sigma$, we say that $r_1 < r_2$ iff there exist a relational database instance K, homomorphisms h_1, h_2 and a set S of atoms, such that

1. $K_1 \not\models h_1(r_1)$,
2. $K_1 \overset{r_1, h_1}{\rightarrow} K_2$,
3. $K_1 \cup S \models h_2(r_2)$,
4. $K_2 \cup S \not\models h_2(r_2)$ and
5. $Null(S) \cap (Null(K_2) - Null(K_1)) = \emptyset$ (i.e. S does not contain new null values introduced in K_2).

We say that Σ is *WA-stratified* (*WA-Str*) iff the constraints in every nontrivial strongly connected component of the *firing graph* $\Gamma(\Sigma) = (\Sigma, \{(r_1, r_2) | r_1 < r_2\})$ are weakly acyclic. □

With respect to c-stratification, *WA-Str* also considers in the satisfaction of constraint r_2, in addition to the database K_1, a set of atoms S (cond. (3)) and atoms in S cannot contain null values introduced in the application of the constraint r_1 (cond. (5)). Moreover, since we are considering strongly connected components (instead of cycles) these components must not be trivial, that is they must have at least one edge, otherwise the constraint cannot be fired cyclically. As a further important observation, in the above definition we consider standard chase for both constructing the graph $\Gamma(\Sigma)$ and checking weak acyclicity.

Example 2. Consider again the set of constraints Σ_1 of Example 1. It is easy to see that, by considering standard chase, does not exist an initial database instance such that the constraint can fire itself, while, by considering the oblivious chase, the constraint fires itself ad infinitum. Thus, the set of constraints Σ_1 is WA-stratified, but not c-stratified. □

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

The following proposition states that *WA-Str* criterion is more general than *CStr* and is not comparable with *SC*. Consequently it is not comparable even with $\mathcal{SwA}$ as $\mathcal{SC}$ is strictly contained in $\mathcal{SwA}$ and $\mathcal{CS}tr$ is not comparable with $\mathcal{SwA}$.

Proposition 1. $\mathcal{CS}tr \subsetneq \mathcal{WA}\text{-}\mathcal{S}tr$ and $\mathcal{SC} \nparallel \mathcal{WA}\text{-}\mathcal{S}tr$. □

It is important to observe that *WA-Str* criterion could be improved by testing safety instead of weak acyclicity over the firing graph. Further improvements could be obtained by considering super-weak acyclicity instead of safety.

Definition 5 (SC-Stratification and SwA-Stratification). Given a set of TDGs Σ, we say that (1) Σ is *SC-stratified* (*SC-Str*) if the constraints in every strongly connected component of the firing graph $\Gamma(\Sigma)$ are safe, and (2) Σ is *SwA-stratified* (*SwA-Str*) if the constraints in every strongly connected component of the firing graph $\Gamma(\Sigma)$ are super-weak acyclic. □

We now analyze the complexity of the above criteria starting by defining a bound on the complexity of the firing problem, i.e. the complexity of checking whether $r_1 < r_2$.

Lemma 1. Let $r_1 : \phi_1 \rightarrow A_1 \wedge \cdots A_k$ and $r_2 : B_1 \wedge \cdots B_n \rightarrow \psi_2$ be two TGDs. The problem of checking whether $r_1 < r_2$ is bounded by $O((k+1)^n)$. □

Although the theoretical complexity of the "firing" problem is exponential, in most cases it is very low (e.g. inclusion dependencies, multivalued dependencies [13]), as usually the number n of body atoms in the fired constraint r_2 is small and the number of atoms in the head of constraint r_1 which could be used to fire r_2 through their unification with B_i (i.e. $k_i > 1$) is even smaller. Indeed, if the number of atoms in the body of r_2 is bounded by a constant, the firing problem is in PTIME. Significative subclasses of constraints for which the firing problem becomes polynomial could be identified, but this is not the aim of this paper.

In the following, for a given set of constraints Σ, we shall denote with C_{ij} the complexity of the problem of checking whether $r_i < r_j$, for $r_i, r_j \in \Sigma$, and with $C_m = max\{C_{ij} | r_i, r_j \in \Sigma\}$.

Proposition 2. Let Σ be a set of TGDs, D be a database Then:

- the problem of checking whether Σ is WA-stratified (resp. SC-stratified, SwA-stratified) is bounded by $O(C_m \times |\Sigma|^2)$;
- if Σ is WA-stratified (resp. SC-stratified, SwA-stratified), the length of every chase sequence of Σ over D is polynomial in the size of D. □

The class of constraints satisfying criterion *C-Str*, for $C \in \{WA, SC, SwA\}$, will be denoted by $\mathcal{C}\text{-}\mathcal{S}tr$. The next theorem states the relationships among the above mentioned criteria and other previously defined conditions.

Theorem 1.

1. $\mathcal{WA}\text{-}\mathcal{S}tr \subsetneq \mathcal{SC}\text{-}\mathcal{S}tr \subsetneq \mathcal{SwA}\text{-}\mathcal{S}tr$,
2. for $C \in \{WA, SC, SwA\}$, $\mathcal{C} \subsetneq \mathcal{C}\text{-}\mathcal{S}tr$ and
3. $\mathcal{SR} \nparallel \mathcal{SwA}\text{-}\mathcal{S}tr$ and $\mathcal{IR} \nparallel \mathcal{SwA}\text{-}\mathcal{S}tr$. □

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

4 Local Stratification

It is trivial that more powerful criteria could be defined by composing criteria which are not comparable. We next present a different generalization of super-weak acyclicity which also generalizes the class $\mathcal{IR}$.

We start by introducing a notion of *fireable place*. We say that a place q appearing in the body of constraint r could be fired by a place q' appearing in the head of constraint r', denoted by $q' < q$, if $q \sim q'$ and $r' < r$. Given two sets of places Q and Q' we say that Q could be fired by Q', denoted by $Q' < Q$ iff for all $q \in Q$ there exists some $q' \in Q'$ such that $q' < q$.

Given a set Q of places, we define $MOVE(\Sigma, Q)$ as the smallest set of places Q' such that $Q \subseteq Q'$, and for every constraint $r = B_r \rightarrow H_r$ in $sk(\Sigma)$ and every variable x, if $Q' < \Pi_x(B_r)$ then $\Pi_x(H_r) \subseteq Q'$, where $\Pi_x(B_r)$ and $\Pi_x(H_r)$ denote the sets of places in B_r and H_r where x occurs.

With respect to the function $Move$, the new function $MOVE$ here considered takes into account the firing of places and not only the unification of places.

Definition 6 (Local Stratification). Given a set Σ of TGDs and two TGDs $r_1, r_2 \in \Sigma$, we say that r_1 triggers r_2 in Σ and write $r_1 \hookrightarrow r_2$ iff there exists an existential variable y in the head of r_1, and a universal variable x occurring both in the body and head of r_2 such that $MOVE(\Sigma, Out(r_1, y)) < In(r_2, x)$. A set of constraints Σ is *locally stratified* (LS) iff the trigger graph $\Delta(\Sigma) = \{(r_1, r_2) | r_1 \hookrightarrow r_2\}$ is acyclic. □

Proposition 3. For every set of TGDs Σ and for every database D
- the problem of checking whether Σ is locally stratified is bounded by $O(C_m \times |\Sigma|^2)$;
- if Σ is locally stratified, the length of every chase sequence of Σ over D is polynomial in the size of D. □

The below theorem states that the class of locally stratified constraints (denoted by $\mathcal{LS}$) is more general than $\mathcal{SwA}$-$\mathcal{S}tr$ and $\mathcal{IR}$.

Theorem 2. $\mathcal{SwA}$-$\mathcal{S}tr \subsetneq \mathcal{LS}$ and $\mathcal{IR} \subsetneq \mathcal{LS}$. □

The next example shows that the containment of $\mathcal{SwA}$-$\mathcal{S}tr \cup \mathcal{IR}$ in $\mathcal{LS}$ is strict.

Example 3. The following set of constraints Σ_3 is locally stratified, but it is neither super-weakly acyclic nor inductively restricted:

$$r_1 : \; N(x) \rightarrow \exists\, y \exists z \; E(x, y) \wedge S(z, y)$$
$$r_2 : \; E(x, y) \wedge S(x, y) \rightarrow N(y)$$
$$r_3 : \; E(x, y) \rightarrow E(y, x)$$

Considering SwA, we have that $Move(\Sigma, Out(r_1, y)) = \{p_3, p_5, p_{10}, p_{13}, p_2, p_{14}\}$ and $In(r_1, x) = \{p_1\} \sqsubseteq Move(\Sigma, Out(r_1, y))$. The trigger graph is cyclic as $r_1 \rightsquigarrow r_1$ and, therefore, Σ_3 is not super-weakly acyclic. As $r_1 < r_3 < r_2 < r_1$ we have that Σ_3 is not SwA-Str as well. Σ_3 is not IR as $r_1 \prec_c r_3 \prec_c r_2 \prec_c r_1$ and for each pair of constraints r_i, r_j such that $r_i \prec_c r_j$, it is possible to construct a database containing null values in positions E_1, E_2, N_1 and S_2 such that whenever r_i fires r_j a null value is propagated from the head of r_i to the head of r_j.
Moreover, as $r_1 \not\prec r_1$ ($MOVE(\Sigma, (r_1, y)) = \{p_3, p_5, p_{13}\}$ and $In(r_1, x) = \{p_1\} \sqsubseteq MOVE(\Sigma, Out(r_1, y))$), $\Delta(\Sigma_3)$ is acyclic and, thus, Σ_3 is locally stratified. □

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

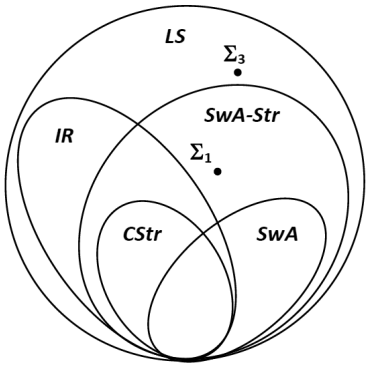


Fig. 1. Criteria Relationships.

5 Conclusions

In this paper we have proposed new criteria for checking chase termination on the base of constraints structural properties. We have shown that the local stratification criterium here introduced, strictly generalize criteria previously proposed in the literature. The relationships among previous criteria and the ones here proposed are reported in Figure 1.

References

1. L. E. Bertossi, "Consistent query answering in databases," *SIGMOD Record*, vol. 35, no. 2, 2006.
2. J. Chomicki, "Consistent query answering: Five easy pieces," in *ICDT*, 2007.
3. G. DeGiacomo, D. Lembo, M. Lenzerini, and R. Rosati, "On reconciling data exchange, data integration, and peer data management," in *PODS*, 2007.
4. R. Fagin, P. G. Kolaitis, and L. Popa, "Data exchange: getting to the core," *ACM Trans. Database Syst.*, vol. 30, no. 1, 2005.
5. P. G. Kolaitis, J. Panttaja, and W. C. Tan, "The complexity of data exchange," in *PODS*, 2006.
6. M. Lenzerini, "Data integration: A theoretical perspective," in *PODS*, 2002.
7. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa, "Data exchange: semantics and query answering," *Theor. Comput. Sci.*, vol. 336, no. 1, 2005.
8. M. Meier, M. Schmidt, and G. Lausen, "On chase termination beyond stratification," *PVLDB*, vol. 2, no. 1, 2009.
9. M. Meier, M. Schmidt, and G. Lausen, "On chase termination beyond stratification," *CoRR*, vol. abs/0906.4228, 2009.
10. B. Marnette, "Generalized schema-mappings: from termination to tractability," in *PODS*, 2009.
11. A. Deutsch, A. Nash, and J. B. Remmel, "The chase revisited," in *PODS*, 2008.
12. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2001.
13. S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*. Addison-Wesley, 1995.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Approximation of the Gradient of the Error Probability for Vector Quantizers

Claudia Diamantini, Laura Genga, and Domenico Potena

Dipartimento di Ingegneria dell'Informazione
Università Politecnica delle Marche - Ancona
{diamantini, genga, potena}@dii.univpm.it

Abstract. Vector Quantizers (VQ) can be exploited for classification. In particular the gradient of the error probability performed by a VQ with respect to the position of its code vectors can be formally derived, hence the optimum VQ can be theoretically found. Unfortunately, this equation is of limited use in practice, since it relies on the knowledge of the class conditional probability distributions. In order to apply the method to real problems where distributions are unknown, a stochastic approximation has been previously proposed to derive a practical learning algorithm. In this paper we relax some of the assumptions underlying the original proposal and study the advantages of the resulting algorithm by both synthetic and real case studies.

1 Introduction

Vector Quantization is the process of mapping continuous or discrete vectors produced by a source, into a finite set of symbols, called the codebook, that can be transmitted or stored using a finite number of bits [1]. Because of its strong mathematical background, accompanied with simplicity and easy of use, Vector Quantization finds application in diverse domains like signal detection, image and video compression, image segmentation, face recognition, speech recognition and clustering, among others. Since introduction in the '70s, researchers continue to study mathematical properties and variants of Vector Quantization, its application to novel data structures, domains and problems [2,3] and to develop learning algorithms for the effective design of the codebook [4,5,6], or to improve efficiency [7]. In the application of Vector Quantization to the classification problem, most of the techniques adopt loss functions different from the misclassification risk to define the learning algorithm, the main reason being that the traditional empirical estimate of the risk is not differentiable. An exception is [8], where a continuous stochastic estimate is exploited to derive the gradient with respect to the codebook. However, in order to be applicable, the general theoretical form of the gradient has been approximated, giving rise to a simple and cheap algorithm called BVQ. This algorithm compares favorably with state of the

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

art techniques [9], but it shows some limitations on high-dimensional spaces and on discrete data. Arguing that this behavior is related to the bias introduced by the approximation when data are sparse, in this paper we present a variant of the algorithm that exploits a finer approximation, improving performance.

In the next section we first present the details about Vector Quantizers architectures used for classification and describe the essential steps that lead to the derivation of the BVQ algorithm. Then the new approximation and solutions to deal with technical issues related to it are presented. In section 3 we show experiments on artificial and real datasets that analyze the behavior and performances of the two algorithms. Finally in Section 4 we draw some conclusions and future work.

2 Bayes Vector Quantizer

This section is devoted to illustrate the theoretical principles underlying the Bayes Vector Quantizer (BVQ) algorithm, a stochastic gradient descent algorithm aimed at minimizing the average misclassification risk performed by a Labeled Vector Quantizer (LVQ). In particular, in the following we start describing the basics of statistical decision theory.

Let boldface characters denote random variables, and let (**x**, **c**) be a random variable pair taking values from $\mathbb{R}^n \times C$, where the continuous random vector **x** is the observation (or feature) vector, while the discrete random variable $\mathbf{c} \in C = \{c_1, c_2, \ldots, c_C\}$ is its class. Classes are statistically characterized by conditional probability density functions (cpdf) $p_{\mathbf{x}|\mathbf{c}}(x|c_i)$, measuring the probability that $\mathbf{x} = x$ given that the class observed is $\mathbf{c} = c_i$. Also, a priori probabilities $P_{\mathbf{c}}(c_i)$, i =1, …, C are given. From cpdf, a-posteriori probabilities class can be derived by the Bayes theorem:

$$P_{\mathbf{c}|\mathbf{x}}(c_i \mid x) = \frac{p_{\mathbf{x}|\mathbf{c}}(x \mid c_i) \cdot P_{\mathbf{c}}(c_i)}{p_{\mathbf{x}}(x)} \tag{1}$$

where $p_{\mathbf{x}}(x)$ is the probability density function of the random vector **x**.

In order to ease notation, boldface subscripts will be dropped when this will not make confusion among random variables.

On the basis of the a-posteriori probabilities in (1), it is possible to introduce a *decision rule* (or *classification rule*), that is a mapping $\Phi(\mathbf{x})$: $\mathbb{R}^n \rightarrow C$, expressing the decision taken to classify the sample **x** being observed. A sample **x** is assigned to the class c_i, for which $p(x|c_i)$ is maximum. The decision rule partitions the observation space into as many decision regions as the classes in C. To evaluate the quality of this decision rule we can resort to the average misclassification risk (or, in short, average risk), that is defined as

$$\int R(\Phi(x) \mid x) p(x) dVx \tag{2}$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

where $R(\Phi(x)|x)$ is defined as *conditional risk*, that is the risk, or cost, of choosing a given class following the analysis of a given sample, and is expressed by

$$R(c_i \mid x) = \sum_{j=1}^{c} b(c_j \mapsto c_i) P(c_j \mid x) \tag{3}$$

where $b(c_j \mapsto c_i)$ are the elements of a cost matrix B, representing the cost of deciding in favor of class c_i when the true class is c_j. If $b(c_j \mapsto c_i) = 1$ for i ≠ j and $b(c_j \mapsto c_i) = 0$ for i = j, then average risk turns to the well-known 0-1 loss, or error probability.

The *Bayes rule* is the classification rule Φ_B that minimizes the average risk.

Before describing the BVQ algorithm, we briefly introduce the main concepts of the model underlying it, namely Labeled Vector Quantizer. A *nearest neighbor Vector Quantizer with Euclidean distance* (VQ) is a mapping $\Omega: \mathbb{R}^n \longrightarrow \mathcal{M}$ where $\mathcal{M}=\{m_1, m_2,..., m_M\}$ is called *codebook* and vectors m_i are called *code vectors*. The relation Ω partitions the $\mathbb{R}^n$ space into M Voronoi regions, where the i-th region is defined as

$$V_i = \left\{ x \in \mathbb{R}^n : \|x - m_i\|^2 \leq \left\|x - m_j\right\|^2 \forall\, j \neq i \right\} \tag{4}$$

The set of aforementioned regions defines the n-dimensional Voronoi diagram of the codebook $\mathcal{M}$.

A *Labeled Vector Quantizer (LVQ)* is a VQ equipped by a further mapping which assigns a label from *C* to each code vector. The decision border of an LVQ is piecewise linear. Having a classification rule based on a LVQ, the equation (2) becomes:

$$R(\Lambda(\Omega)) = \sum_{j=1}^{C} \sum_{i=1}^{M} b(c_j \mapsto l_i) \int_{V_i} P(c_j \mid x) p(x) dV_x \tag{5}$$

where $l_i \in C$ is the label of the code vector m_i. Note that the average risk depends on Voronoi regions, which in turn depends on the mutual position of code vectors; thus, it is possible to modify the average risk by modifying their positions. A principled way to do this is to adopt gradient descent techniques:

$$\begin{aligned} & m_i^{(k+1)} = m_i^{(k)} - \gamma^{(k)} \nabla_i R^{(k)}(\Phi^{(k)}) \\ & i = 1,2,\ldots,M; \quad k = 0,1,.... \end{aligned} \tag{6}$$

where $m_i^{(k)}$ is the code vector m_i at the k-th iteration of the algorithm, $\gamma^{(k)}$ is the step size and $\nabla_i R(\Omega)$ is the i-th component of the gradient of the average risk. At each iteration of the algorithm the code vectors are updated (and hence the decisional rule) on the basis of the k-th measurement of the average risk gradient.

To compute the gradient, it is important to note that, when *p(x)* is smooth, the average risk is differentiable with respect to the position of code vectors, hence the analyt-

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

ical form of the gradient can be obtained [8]. The gradient with respect to the code vector m_i takes the form:

$$\nabla_i R(\Omega) = \sum_{j=1}^{C} \sum_{q=q_1}^{q_N} \frac{b(c_j \mapsto l_q) - b(c_j \mapsto l_i)}{\|m_i - m_q\|} \times \int_{S_{i,q}} (m_i - x) p(c_j \mid x) p(x) dS_x \quad (7)$$

where $q_1 \ldots q_n$ are the indexes of code vectors that are adjacent to m_i, and $S_{i,q}$ is the decisional surface (i.e. a (n-1) dimensional hyperplane) separating Voronoi regions of m_i and m_q. In order to obtain the gradient when class distribution are unknown (as usual in real classification problems), we refer to the Parzen estimate [10, Chap. 6]. Hence the gradient becomes:

$$\nabla_i \boldsymbol{R}(\Omega) = \sum_{q=q_1}^{q_N} \frac{b(\boldsymbol{h} \mapsto l_q) - b(\boldsymbol{h} \mapsto l_i)}{\|m_i - m_q\|} \times \int_{S_{i,q}} (m_i - x) w(x - \boldsymbol{z}) dS_x \quad (8)$$

where $w(x)$ is the kernel or Parzen window, $\mathbf{z}$ is a sample randomly chosen from the training set and $\mathbf{h}$ is its class label. The derivation of the gradient formula is not reported due to page limitation. Interested readers can refer to [8].

The Bayes Vector Quantizer algorithm uses a particular form of $w(x)$ that allows a simple and cheap implementation. In particular, we consider a hypercubic window, that is $w(x) = \Delta^{-n}$ over a n-dimensional hypercube of side Δ centered on the origin and $w(x) = 0$ elsewhere. In such a way, the integral in (8) assumes a non-zero value for each point on the decision border which is less than $\Delta/2$ from the projection of $\mathbf{z}$ over the border; in other words $\nabla_i R(\Omega) \neq 0$ for all the pairs m_i, m_q such that $S_{i,q}$ falls into the window centered in $\mathbf{z}$ (i.e. $\exists x \in S_{i,q}: w(x - \boldsymbol{z}) = \Delta^{-n}$). At each iteration of the algorithm, only code vectors in these pairs are updated, while other code vectors remain unchanged.

The main issue is to check whether the piece of decision border $S_{i,q}$ falls in the window. In order to solve this issue, we could compute the n-dimensional Voronoi diagram or its dual, namely the Delaunay diagram [11, 12]. Although the use of these diagrams allows a precise calculation of the gradient, their definitions makes the algorithm highly inefficient. Furthermore, experimental results have not shown a significant improvement in performance with respect to implementations that we will present in this work. Hence, we left this direction in place of approximate solutions. To this end, note that if we consider only the piece of border nearest to $\mathbf{z}$ (i.e. the one related to the first two code vectors nearest to $\mathbf{z}$), the issue of checking whether $S_{i,q}$ falls in the window becomes to check whether the distance between the point $\mathbf{z}$ and the line $S_{i,q}$ is less than $\Delta/2$. Hence, the BVQ2 version has been introduced, where at each iteration only the two nearest code vectors are updated [8].

In this version the formula (8) becomes:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

$$\begin{cases} \nabla_1 R(\Phi) = \dfrac{(b(\boldsymbol{h} \mapsto l_2) - b(\boldsymbol{h} \mapsto l_1)) \cdot (m_1 - z_{1,2})}{\Delta \| m_1 - m_2 \|} \\ \nabla_2 R(\Phi) = \dfrac{(b(\boldsymbol{h} \mapsto l_1) - b(\boldsymbol{h} \mapsto l_2)) \cdot (m_2 - z_{1,2})}{\Delta \| m_1 - m_2 \|} \end{cases} \tag{9}$$

where $\mathbf{z}_{1,2}$ is the projection of the point $\mathbf{z}$ on the surface $S_{1,2}$ separating the first two code vectors nearest to $\mathbf{z}$, namely m_1 and m_2. The assumption underlying BVQ2 is that the window intersects only one piece of the decisional border at a time. This assumption can be considered true since the unbiasedness of Parzen estimate requires that the span of the window is reduced as the training set grows. BVQ2 has experimentally shown comparable or better results than those obtained by well-known and effective classification algorithms; furthermore, from computational point of view BVQ2 turns out to be an advantageous choice [8, 9].

When the training sample falls close to a vertex of the Voronoi diagram, that is the window intersects more than one piece of the decision border, then the approximation error introduced in (9) becomes not negligible.

In these cases we can improve the classifier performances by updating more than two code vectors, achieving a better approximation of (8). In particular, in this work we introduce BVQ3, a version of the BVQ algorithm which updates up to 3 code vectors at each iteration. In this case, the components of the gradient, for the first three code vectors nearest to $\mathbf{z}$ (m_1, m_2 and m_3 respectively), are:

$$\begin{cases} \nabla_1 R(\Phi) = \dfrac{(b(\boldsymbol{h} \mapsto l_2) - b(\boldsymbol{h} \mapsto l_1)) \cdot (m_1 - z_{1,2})}{2\Delta \| m_1 - m_2 \|} + \dfrac{(b(\boldsymbol{h} \mapsto l_3) - b(\boldsymbol{h} \mapsto l_1)) \cdot (m_1 - z_{1,3})}{2\Delta \| m_1 - m_3 \|} \\ \nabla_2 R(\Phi) = \dfrac{(b(\boldsymbol{h} \mapsto l_1) - b(\boldsymbol{h} \mapsto l_2)) \cdot (m_2 - z_{1,2})}{2\Delta \| m_1 - m_2 \|} + \dfrac{(b(\boldsymbol{h} \mapsto l_3) - b(\boldsymbol{h} \mapsto l_2)) \cdot (m_2 - z_{2,3})}{2\Delta \| m_2 - m_3 \|} \\ \nabla_3 R(\Phi) = \dfrac{(b(\boldsymbol{h} \mapsto l_1) - b(\boldsymbol{h} \mapsto l_3)) \cdot (m_3 - z_{1,3})}{2\Delta \| m_1 - m_3 \|} + \dfrac{(b(\boldsymbol{h} \mapsto l_2) - b(\boldsymbol{h} \mapsto l_3)) \cdot (m_3 - z_{2,3})}{2\Delta \| m_2 - m_3 \|} \end{cases} \tag{10}$$

The pseudo code of the BVQ3 version is shown in Figure 1. Again, we have to check whether the intersection between the decision border at each iteration and the hypercubic window is not null. To this end, first we check whether the piece of decision border formed by the first two code vectors close to $\mathbf{z}$ are in the window (step 4.3). This step is the same as in BVQ2 version. Please note that, if these two code vectors belong to the same class, then the surface dividing their Voronoi regions is not a piece of decision border. In this case, whatever the class $\mathbf{h}$ of $\mathbf{z}$, it turns out that $b(\boldsymbol{h} \mapsto l_1) = b(\boldsymbol{h} \mapsto l_2)$, making zero right sides of (9); and, the two code vectors are de-facto not updated. Hence, in order to speed up the execution avoiding an unnecessary check, (at the step 4.3) we introduced also the check on the classes of m_1 and m_2.

Note also that if the piece of border between m_1 and m_2 does not intersect the window, then surely no other piece of border can intersect it. Therefore, if the check in 4.3 is false, no other check is needed and the algorithm goes to next iteration. Otherwise, we have to check whether pieces of decision border formed by m_3 fall in the window, as in Figure 2.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

To this end, before all, we have to check whether the third code vector forms borders with m_1 or m_2, that is to check whether the Voronoi region of m_3 is adjacent to at least one of the region of m_1 and m_2. This further checking is needed due the fact that the existence of a Voronoi surface between the first and the third code vector or between the second and the third one is not guaranteed, while it always exists between the first two nearest code vectors. In this case, we could again use the analytical form of the Voronoi diagram, but as said above its computation is inefficient.

Let label(**x**) be the function that returns the class of **x**.
1. Set the value of Δ, γ^0, the number of iterations n_{max} and the maximum number of attempts *try_max*;
2. Initialize code vectors m_i.....m_n;
3. Set $flag_{BVQ3} = 0$;
4. For k = 1 to n_{max} do
   1. Randomly pick a training pair ($z^{(k)}$, $h^{(k)}$) from the training set;
   2. Find the first three code vectors m_1, m_2 ,m_3 nearest to $z^{(k)}$, such that
      $$\left\|m_1 - z^{(k)}\right\|^2 \leq \left\|m_2 - z^{(k)}\right\|^2 \leq \left\|m_3 - z^{(k)}\right\|^2$$
   3. If (label(m_1) ≠ label(m_2) and $\left\|z^{(k)} - z^{(k)}_{1,2}\right\| \leq \Delta/2$)
      1. If (label(m_1) ≠ label(m_3) or label(m_2) ≠ label(m_3))
         1. Set *try=0;*
         2. While ($flag_{BVQ3}=0$ and *try* < *try_max*)
            1. Set $d=z^{(k)}$ + *noise;*
            2. *try* = *try* + *1;*
            3. If $w(d - z^{(k)}) \neq 0$
               1. Find the first 3 code vectors q_1,q_2,q_3 nearest to d, such that: $\|q_1 - d\|^2 < \|q_2 - d\|^2 < \|q_3 - d\|^2$
               2. If ($q_1 = m_3$ and ($q_2 = m_1$ or $q_2 = m_2$))
                  1. update m_1, m_2 ,m_3 using the formula in (10);
                  2. $flag_{BVQ3} = 1$;
      2. else update m_1 and m_2 using the formula in (9);

Fig. 1. Pseudo-code of BVQ3 algorithm

Hence, in order to reduce complexity, the adjacency of Voronoi regions is empirically tested according to the following principle: regions V_1 (or V_2) and V_3 are adjacent if a point **d** closed to **z** exists, such that m_3 and m_1 (or m_2) are the two code vectors nearest to **d**. To implement this idea, we add a Gaussian n-dimensional noise to the training sample **z**, and then we find the neighborhood of the new point **d** = **z** + **noise** (see Figure 2). The variance of the noise has to be set to a value smaller than the size of the window. As a matter of fact, values comparable to (or bigger than) Δ would have the effect of altering the learning of BVQ, as if de-facto we extend the size of Δ. Now, if **d** falls in the window (step 4.3.1.2.3) and the adjacency of V_3 and V_1 (or V_2) is established (step 4.3.1.2.3.2), then m_1, m_2 and m_3 are updated according

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

to (10). The checking of adjacency is repeated at most *try_max* times. If the adjacency cannot be established, then only m_1 and m_2 are updated using the formula (9), as in BVQ2.

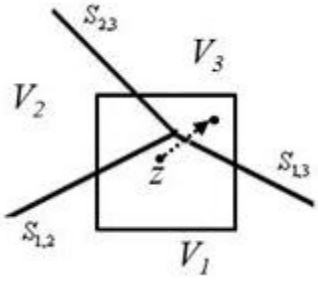


Fig. 2. Example of a point who satisfies the update conditions

We like to highlight that although the proposed BVQ3 version is, of course, more time-consuming than BVQ2, the two version have the same complexity order. As a matter of fact, the main differences are in the extraction of three neighbors instead of two, and in the checking the adjacency of third region. However, whatever the number of nearest code vectors to find, it is needed to order all code vectors. The latter check implies the computation of $^{try\_max}/_{2}$ distances on average. In the next Section, we experimentally show the advantages of BVQ3 over BVQ2.

3 Experiments

This section is devoted to experimentally evaluate the BVQ3 algorithm and to compare it with the BVQ2 version. We first examine their accuracy and ability to approximate the Bayes decision border on a synthetic experiment. Then we show the performances on real-world datasets from the UCI Machine Learning Repository [13].

In both cases the figure of merit to compare the two algorithms is the error probability, hence the cost matrix has been always set to [0 1;1 0]. In particular, for each dataset we show results obtained by using a 10-fold cross validation procedure. The same folds are used both for BVQ2 and BVQ3.

3.1 Experiment with a synthetic dataset

The first experiment has been performed on a two classes 2-D dataset. Each class has been drawn from 2-D independent Gaussian distributions, with the same a-priori probability and zero means. The two classes can be distinguished by means of their covariance matrixes, I for class c1 and 0.01·I for class c2 respectively, where I is the 2×2 identity matrix.

In order to determine the best Δ window size, we refer to the Parzen method to derive the variance that minimizes the error probability on the test set. In particular, by

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

centering a Gaussian function on each training sample, an estimated probability density function (pdf) has been obtained for both class. Based on the estimated pdfs and the classes' a priori probability, test samples have been classified according to the Bayes decision rule. Varying the variance of the Gaussian distribution we are able to obtain different estimate of data distribution and, hence, classification. The estimate that minimizes the classification error is the best approximation of the data distribution, and returns the optimal variance. For this dataset, the best value of the variance is σ^2=2.1·10-5. As in BVQ we use a hypercubic window instead of a Gaussian distribution, the value of delta is given by the following formula: $\Delta = \sqrt{12 \cdot \sigma^2} = 0.0159$.

The number of iteration was fixed at 40000, and several experiments have been performed, by varying the γ^0 value, the number of code vectors and, in the case of BVQ3, the *try_max* value. In particular, we set γ^0 values within the interval [0.00159, 0.0159] with a step of 0.02. Every experiments has been repeated by doubling the number of code vectors, ranging from 4 to 128. The *try_max* parameter has been set to 10,20 and 50.

Table 1 reports best results with respect to the number of code vectors, obtained by varying other parameters. Values are averages resulting by applying a 10-fold cross validation procedure, bold ones are the minimal errors for the two BVQ versions. In both versions, the best accuracy is achieved with 64 code vectors, where the BVQ3 outperforms the result of BVQ2. We can note that the result are close to the Bayesan error, which for this problem is 0.027. In general BVQ3 provides better or at least comparable results than BVQ2.

Table 1. Error probability on *Gaussian*

	Number of code vectors					
	4	8	16	32	64	128
BVQ2	0.0425	0.0295	0.0295	0.0285	**0.0282**	0.029
BVQ3	0.0415	0.0296	0.0291	0.0289	**0.0277**	0.0288

In order to evaluate the ability of approximating the Bayes decision we report the initial (Figure 3.(a)) and the final (Figure 3.(b)-(c)) code vector configurations reached by the two BVQ versions, corresponding to the minimum error values obtained in Table 1. In the Figure, points are code vectors and lines define the Voronoi regions of any code vectors. The bold line is the decision border, while the dotted line represents the optimal (Bayesian) decision border, which for this problem is a circle. Code vectors in the initial configuration are randomly chosen.

Notice that the 3-vector updating determines a better approximation of the border than BVQ2. Furthermore, the distribution of code vectors in BVQ3 is more regular than in the previous version.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

3.2 Experiments with real-world datasets

As mentioned before, in this section we analyze the results of the experiments conducted on 4 datasets chosen from the UCI Machine Learning repository, namely Australian, Liver, Mushroom and Ionosphere datasets. Main characteristics of these datasets are reported in Table 2.

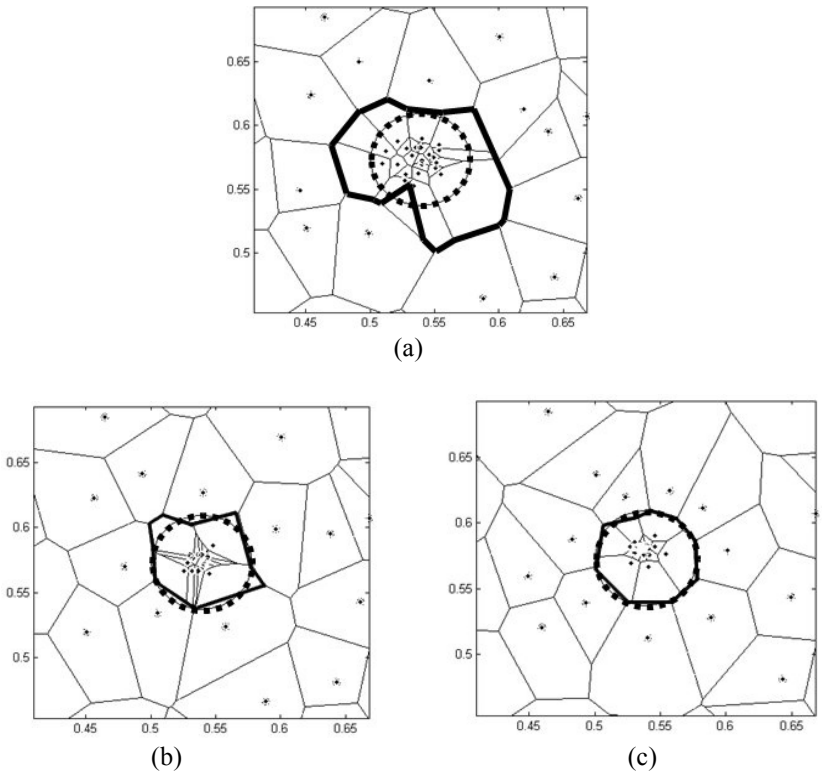


Fig. 3. Example of convergence of the algorithm for *Gaussian*: (a) Initial code-vector layout, (b) final BVQ2 configuration, and (c) final BVQ3 configuration.

As a first step, a data normalization procedure has been performed, with the aim to have all features within the [0, 1] interval, so to give equal importance to each feature during learning. To this end, we have applied the following formula:

$$x_i' = \frac{x_i - \min_i}{\max_i - \min_i} \tag{11}$$

where x_i is the original feature and x_i' is the normalized one, $\min_i$ and $\max_i$ are minimum and maximum values of the feature x_i respectively.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Table 2 reports the dataset characteristics. As for the previous experiment we use the Parzen method to calculate the Δ window size. The number of iterations has been fix to 50000, γ^0 assumes values in [0.1, 1], with a step of 0.1, and the number of code vectors in the set {4, 8, 16, 32, 64, 128}. For BVQ3, we set the *try_max* parameter in the set [10,20,30].

Table 2. Dataset characteristics: number of features (dim), number of samples (N), number of samples in each class (N_1 and N_2). Δ is the window size estimated through the Parzen method.

Dataset	dim	N	N_1	N_2	Δ
Australian	14	690	307	383	0.346
Liver	6	345	145	200	0.154
Mushroom	22	8124	4208	3916	1.897
Ionosphere	34	351	126	225	1.897

Table 3. Error probability on each *real-world dataset*

		Number of code vectors					
Dataset		4	8	16	32	64	128
Australian	BVQ2	0.1406	0.1420	0.1478	0.1420	**0.1391**	0.1522
	BVQ3	0.1478	**0.1319**	0.1449	0.1478	0.1406	0.1391
Liver	BVQ2	0.3205	0.3200	**0.3190**	0.3505	0.3505	0.3700
	BVQ3	0.3267	**0.2919**	0.2981	0.3305	0.3324	0.3348
Mushroom	BVQ2	0.1001	0.0617	0.0327	0.0211	0.0113	**0.0065**
	BVQ3	0.0996	0.0373	0.0201	0.0049	**0.0017**	0.0046
Ionosphere	BVQ2	0.1480	0.1167	0.1082	0.1254	**0.0998**	0.1111
	BVQ3	0.1426	0.1194	0.0944	0.0898	0.0944	**0.0639**

Table 3 summarizes the best results obtained in terms of average error with varying the number of code vectors. The best results for each dataset is in bold.

Results in the table confirm the results of experiments with the synthetic dataset, namely BVQ3 outperforms results of the BVQ2 version. As a matter of fact, BVQ3 returns best results in each dataset. The improvement is particularly significant in Mushroom and Ionosphere, where the gain of BVQ3 over BVQ2 is more than 36%.

Analyzing the results obtained in further detail, it turns out that in general the three-vector version performs better for all *try_max* configurations. It is however not possible to identify the best absolute configuration, as the best minimum error values are achieved for different values of *try_max* in different datasets. In particular, the best value of *try_max* is 10 for the datasets Liver and Ionosphere, 30 for Australian and Mushroom.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Concerning the computing time BVQ3 is, as said before, more time-consuming than BVQ2 : however the difference between the computing times of the two algorithms is not particularly significant, compared to the considerable improvement of classification performance achieved by BVQ3. As an example we reports in Figure 4 the trend of computing times of BVQ2 and BVQ3 calculated on the dataset *Ionosphere*: the continuous line refers to BVQ3, the dotted line to BVQ2.

We can note that two lines have the same trend and are quite close, which corresponds to two similar computing times.

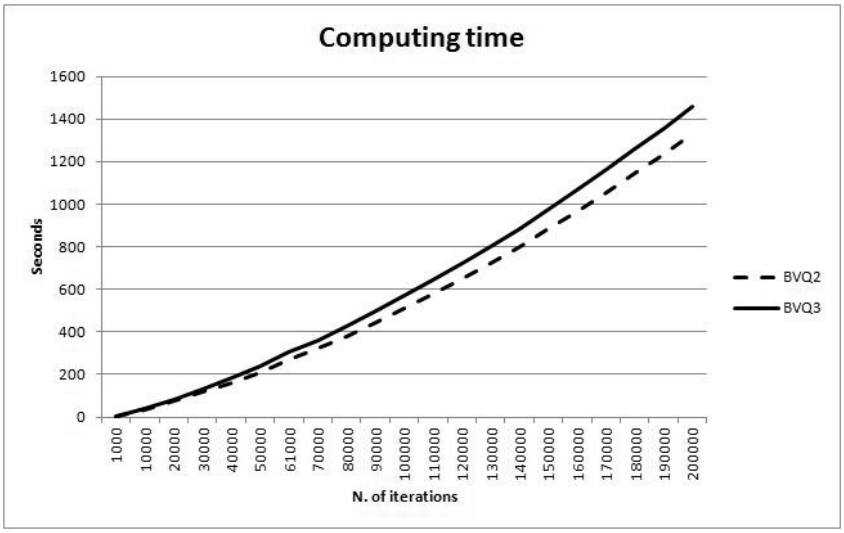


Fig. 4. Comparison between computing time of BVQ3 and BVQ2 for *Ionosphere.*

4 Conclusions

In the paper we presented a new version of the algorithm BVQ, obtained by dropping the hypothesis that at each iteration only a segment of the piece-wise linear border defined by a labeled Vector Quantizer is interested by the updating. Such hypothesis leads to an asymptotically unbiased estimate of the true gradient, since asymptotically the Parzen window involved in the estimate shrinks to a single point. However, in real cases where a finite number of training data is available the hypothesis introduce a bias that worsen the performance of the algorithm. As it is demonstrated in the paper, the recognition of cases where a training data falls close to a vertex of the Voronoi diagram, and the consequent updating of the interested code vector leads to more accurate approximation of the true decision border and to better performance without adding significant complexity.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

We plan to extend the empirical study by performing more experiments on real data, and to compare the new algorithm with other state of the art approaches, also on multi-class, unbalanced datasets.

5 References

1. Gray, R. : Vector quantization. In: Acoustics, Speech, and Signal Processing Magazine, vol. 1, no. 2, (1984).
2. Jain, B.J., Obermayer, K. : Generalized learning graph quantization. In: Proc. of the 8th International Conference on Graph-based Representations in Pattern Recognition, pp. 122-131, Springer-Verlag, Berlin, Heidelberg, (2011).
3. Sanchez, J.; Perronnin, F. : High-dimensional signature compression for large-scale image classification. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1665-1672, (2011)
4. Tsai, C., Lee, C., Chiang, M., Yang, C., : A fast VQ codebook generation algorithm via pattern reduction. In: Pattern Recognition Letters, Vol. 30, no. 7, pp. 653-660, (2009).
5. Lazebnik, S.; Raginsky, M. : Supervised Learning of Quantizer Codebooks by Information Loss Minimization. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, , vol.31, no.7, pp.1294-1309, (2009)
6. Kastner, M., Hammer, B., Biehl, M., Villmann, T. : Functional Relevance Learning in Generalized Learning Vector Quantization. Neurocomputing, in press. Pre-print available at http://www.cs.rug.nl/biehl/Publications/prepnow.html (2012)
7. Kekre, H. B., Sarode, T.K. : Fast codebook search algorithm for vector quantization using sorting technique. In: Proc. of the International Conference on Advances in Computing, Communication and Control, ACM, New York, NY, (2009)
8. Diamantini,C., Spalvieri, A. : Quantizing for Minimum Average Misclassification Risk. In: IEEE Trans. Neural Netw., vol. 9, no. 1, pp. 174–182 (1998).
9. Diamantini, C., Potena, D. :Bayes Vector Quantizer for Class-Imbalance Problem. In : *Knowledge and Data Engineering, IEEE Transactions on* , vol. 21, no. 5, pp.638-651 (2009).
10. Fukunaga,K.: Introduction to Statistical Pattern Recognition (2nd Edition). Academic Press, San Diego (1990).
11. Aurenhammer, F., Klein, R. : Voronoi Diagrams. In: Handbook of Computational Geometry, Sack,J., Urrutia,J., (eds), cap.5, pp.201-290, Elsevier Science Publishers, Amsterdam, (2000).
12. Okabe,A., Boots ,B., Sugihara, K. : Spatial Tessellations - Concepts and Applications of Voronoi Diagrams(2nd Edition). Wiley, Toronto (2000).
13. Blake,C.L., Hettich, S., Merz, C.J., Newman, D.J.: UCI Repository of Machine Learning Databases, http://kdd.ics.uci.edu/, (1998).

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Topic Modeling for Segment-based Documents*

Giovanni Ponti[1], Andrea Tagarelli[2], and George Karypis[3]

[1] ENEA - Portici Research Center, Italy
[2] Department of Electronics, Computer and Systems Sciences, University of Calabria, Italy
[3] Department of Computer Science & Engineering, Digital Technology Center, University of Minnesota, Minneapolis, USA

Abstract. Statistical topic models have traditionally assumed that a document is an indivisible unit for the generative process, which may not be appropriate to handle documents that are relatively long and show an explicit multi-topic structure. In this paper we describe a generative model that exploits a given decomposition of documents in smaller, topically cohesive text units, or segments. The key-idea is to introduce a new variable in the generative process to model the document segments in order to relate the word generation not only to the topics but also to the segments. Moreover, the topic latent variable is directly associated to the segments, rather than to the document as a whole. Experimental results have shown the significance of the proposed model and its better support for the document clustering task compared to other existing generative models.

1 Introduction

In recent years, there has been a growing interest towards *statistical topic models* [1–6], which assume that a document can be represented as a mixture of probability distributions over its constituent terms, where each component of the mixture refers to a main topic. The document representation is obtained by a generative process, i.e., a probabilistic process that expresses document features as being generated by a number of (latent) variables. Unlike conventional vector-space text models, topic models are able to involve latent semantic aspects underlying correlations between words to leverage the structure of topics within a document. This ability becomes particularly relevant when documents explicitly belong to multiple topical classes, and the different topics are discussed at different parts of the text, which is frequent in real-world datasets.

However, classic generative models for documents like PLSA [1] and LDA [2] are not really able to capture topic correlations. A major reason behind this limitation is that they rely on the bag-of-words assumption, which allows for keeping the model's computational complexity acceptable, but also incorrectly assumes independence among the word-topics in the document.

In this work, we present a *Segment-based Generative Model* (SGM) which allows for alleviating the limitations due to the bag-of-words assumption in the context of

* A full version of this paper appeared at the 14th Int. Conf. on Discovery Science (DS), Espoo, Finland (2011)

* A full version of this extended abstract appeared at the 14th Int. Conf. on Discovery Science (DS), Espoo, Finland (2011)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

(multi-topic) documents by exploiting the underlying composition of documents into topically coherent text blocks, or *segments*. Unlike other existing generative models, term generation in SGM is related not only to topics but also to segments. As a consequence, the latent variable that models topics is being associated to the within-document segments, rather than to the document as a whole. In addition, although this model will continue to treat each segment as a bag-of-words, the word-to-topic assignments will be contextualized w.r.t. the various segments, thus generating proper topic distributions for each term according to the segment in which the term occurs.

We evaluated the effectiveness of our generative model in a document clustering task. Experiments conducted on multi-topic document collections have shown that our segment-based approach to document generative modeling improves document clustering performance w.r.t. the other competing models. Moreover, clustering of topically-segmented documents based on our generative model has shown to outperform a traditional document clustering approach in which segments are represented based on the conventional vector-space model.

2 Related Work

The problem of identifying a topic feature space in a given document collection has been originally addressed by mapping the term-document representation to a lower-dimensional latent "semantic" space. Following this line, one of the earliest methods was *Probabilistic Latent Semantic Analysis* (PLSA) [1], in which the conditional probability between documents and terms is modeled as a latent variable. An extension of PLSA, called *Ext-PLSA* [5], has also been proposed to specifically support document clustering. *Latent Dirichlet Allocation* (LDA) [2] is a corpus-oriented model, since the generative process consists of a three-level scheme that involves the whole collection, the documents, and the words in each document. Since exact inference in LDA is not tractable, a number of approximate inference approaches have been developed. Moreover, although possessing a consistent generative semantics, LDA is not able to capture correlations among topics, since the topic proportions as derived from a Dirichlet distribution are substantially independent.

Text segmentation is concerned with the fragmentation of an input text into smaller units (e.g., paragraphs) each possibly discussing a single main topic. Regardless of the presence of logical structure clues in the document, linguistic criteria and statistical similarity measures have been mainly used to identify thematically-coherent, contiguous text blocks in unstructured documents (e.g., [7–9]). The *TextTiling* algorithm [7] is the exemplary similarity-block-based method, which has been successfully used in several application domains for retrieval purposes. TextTiling is able to subdivide a text into multi-paragraph, contiguous and disjoint blocks that represent passages, or subtopics.

To the best of our knowledge, there are only a few studies that address topic modeling and text segmentation in a combined way [10–12]. The key idea is generally to improve the performance of text segmentation algorithms under the assumption that topic segments tend to be lexically cohesive and a switch to a topic corresponds to a shift in the term distribution. Our proposal differs from these methods significantly, since it does not define a new topic-based segmentation approach. Rather, we design a document generative model specifically for topically-segmented documents. Thus, being able to involve terms as well as text segments in a document in the generative

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

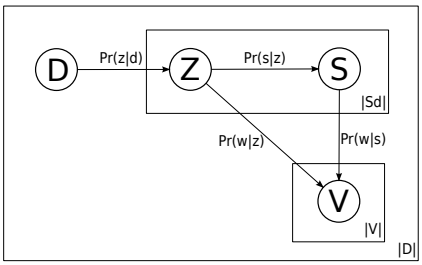


Fig. 1. Plate-based graphical model representation of SGM. The outer plate represents documents (d), whereas the inner plates represent the repeated choice of topics (z) and segments (s) and of words (w) within a document

process, our approach aims to lead to a finer-grained identification of topic distributions in the document generative process.

In the same direction as ours, the STM model [13] also exploits the availability of document segments in the generative process. It substantially extends LDA by introducing a further level to represent the document segments. Although our SGM and STM are both generative models that handle document segments, they are quite different in terms of latent variable dependences (STM is a generative model for a corpus, whereas SGM is able to generate model topics for a single document independently of the others in the collection), and of generative process complexity (a four-level model for STM w.r.t. a two-level one for SGM).

3 Segment-based Generative Model

We are given a collection of documents $\mathcal{D} = \{d_1, \ldots, d_N\}$ and a set of words $\mathcal{V} = \{w_1, \ldots, w_M\}$, which represents the vocabulary of $\mathcal{D}$. Each document $d \in \mathcal{D}$ is a sequence of n_d words. We denote with $\mathcal{Z} = \{z_1, \ldots, z_T\}$ the set of hidden topics, where $\mathcal{Z}$ represents a latent variable model that associates topics (unobserved class variables) with word occurrences (observed data). We suppose that each document $d \in \mathcal{D}$ is provided as a set S_d of contiguous, non-overlapping text blocks, or *segments*, and that such segments are obtained by some text segmentation method (cf. Section 2). However, we do not make any particular assumption about the document segmentation strategy (provided that it is in principle coherent to the topical structure of the documents) and the algorithmic choices of the specific text segmentation method used.

Figure 1 illustrates the graphical model representation of SGM, by which nodes correspond to variables and boxes are *plates* representing replicates of the enclosed variables. SGM utilizes one latent variable $\mathcal{Z}$ to model topic distributions, whereas the model variable $\mathcal{S} = \{S_1, \ldots, S_N\}$ is used to represent document segments. The generative process performed by SGM on a corpus $\mathcal{D}$ of segmented documents can be summarized as follows:

1. Select a document d from $\mathcal{D} \Rightarrow \Pr(d)$
2. For each segment $s \in S_d$:
   a) Choose a topic z for the document $d \Rightarrow \Pr(z|d)$
   b) Associate topic-to-segment probability to the segment s for the selected topic z $\Rightarrow \Pr(s|z)$
   c) For each word w in the segment s:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Table 1. Datasets used in the experiments

dataset	*size (#docs)*	*#words*	*#topic-labels*	*avg #topic-labels per doc*	*#topic-sets*	*avg #docs per topic-set*
IEEE	4,691	129,076	12	4.56	76	61.72
PubMed	3,687	85,771	15	3.20	33	111.73
RCV1	6,588	37,688	23	3.50	49	134.45

– Choose a word w from the current topic and segment $\Rightarrow \Pr(w|z,s)$

The above generative process can be translated into a joint probability model for triadic data, in which each observation is expressed by a triad defined on documents, segments, and words:

$$\Pr(d,s,w) = \Pr(d) \sum_{z \in \mathcal{Z}} \Pr(z|d) \Pr(s|z) \Pr(w|z,s)$$

Model parameter estimation is accomplished by the Expectation Maximization (EM) algorithm. Since SGM has one latent variable ($\mathcal{Z}$) that models the document topics, the E-step consists in estimating the posterior probabilities of $\mathcal{Z}$ given the known model variables:

$$\Pr(z|d,s,w) = \frac{\Pr(z,d,s,w)}{\Pr(d,s,w)} = \frac{\Pr(z|d) \Pr(s|z) \Pr(w|z,s)}{\sum_{z \in \mathcal{Z}} \Pr(z|d) \Pr(s|z) \Pr(w|z,s)}$$

The M-step aims to maximize the expected value of the log-likelihood, $\mathbf{E}[\mathcal{L}]$, which is computed as

$$\mathbf{E}[\mathcal{L}] = \sum_{d \in \mathcal{D}} \sum_{s \in S_d} \sum_{w \in \mathcal{V}} n(d,s,w) \times \sum_{z \in \mathcal{Z}} \Pr(z|d,s,w) \log(\Pr(d,s,w))$$

where $n(d,s,w)$ is the number of occurrences of word w in the segment s of a given document d. Note that the above formula takes into account only the relevant part of the log-likelihood function, since it is trivial to estimate $\Pr(d)$ as proportional to $\sum_{s \in S_d} \sum_{w \in \mathcal{V}} n(d,s,w)$.

4 Evaluation and Results

4.1 Methodology

We conducted an experimental evaluation aimed at assessing the impact of using the SGM representation of documents on the performance of a *document clustering* task. Clustering documents with an inherent multi-topic structure is traditionally accomplished by a *soft* (e.g., fuzzy) clustering method to produce overlapping clusters of documents. However, we pursue the idea that the particular document representation offered by generative models can enable simpler (i.e., *hard*) clustering schemes. Since the generative process produces a topic distribution for each document in the corpus (i.e., $\Pr(z|d)$), documents are represented as *probability mass functions* (pmfs) that are defined over a feature space underlying topics. This topic-feature space is usually lower-dimensional than conventional term-feature space, and is identified by a mixture model of the topic distributions for any given document.

To perform document clustering, we used a centroid-based-linkage agglomerative hierarchical algorithm for clustering document pmfs, which was developed in our earlier work [14]. In the algorithm, the notion of prototype (centroid) of a cluster is defined as a mixture that summarizes the pmfs of the documents within that cluster. Moreover,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

the cluster merging criterion, which decides the pair of clusters to be merged at each step, utilizes the Hellinger distance to compare the cluster prototypes. Note that the Hellinger distance can be viewed as the information-theoretic counterpart of the popular cosine similarity, since it is derived from the Bhattacharyya coefficient [15] which represents the cosine between any two vectors that are composed by the square root of the probabilities of their mixtures.

For the experimental evaluation, we used three collections of multi-topic documents belonging to different application domains, whose main characteristics are summarized in Table 1. To perform document segmentation, we used TextTiling (cf. Section 2) and set its parameters around the values suggested in [7], by varying the token-sequence size around ± 10 of the default 20 and the text unit size from 3 to 15. We finally selected three configurations, corresponding to the minimum, the average, and the maximum segmentation level (i.e., number of segments produced); we will use symbols SGM^{min}, SGM^{avg}, and SGM^{max} to refer to instances of SGM applied to these three segmentation schemes, for a given document collection.

We adopted an external cluster validity approach, in order to assess how well a document clustering solution fits the *topic-set-based reference classification* for a given dataset. We derived such reference classifications by exploiting the availability of topic-labels in each dataset: since topic distributions identify the set of covered topics in each document, documents that are clustered together tend to have similar profiles of their mixtures of topics. We call a *topic-set* θ a subset of topics in $\mathcal{Z}$ that is entirely covered by at least one document. Topic-sets are regarded as sets of topic-labels that may overlap, whereas documents are kept organized in disjoint groups. Therefore, the assignment of topic-sets to documents allows for inducing a multi-topic, hard classification for the documents in a given dataset, which can be exploited as a reference classification for clustering evaluation purposes. The last two columns of Table 1 report on statistics about the topic-sets that were identified on each of the evaluation datasets, with a coverage of at least 20 documents per topic-set. As an example of topic-set construction, consider a set of documents $\mathcal{D} = \{d_1, \ldots, d_7\}$ and a set of topic-labels $\mathcal{Z} = \{z_1, \ldots, z_5\}$ in $\mathcal{D}$. Suppose that an external document labeling information produces an assignment of each document in $\mathcal{D}$ with a subset of topics in $\mathcal{Z}$ as follows: $d_1 \leftarrow \{z_3, z_5\}, d_2 \leftarrow \{z_1, z_4\}, d_3 \leftarrow \{z_1, z_2, z_5\}, d_4 \leftarrow \{z_1, z_4\}, d_5 \leftarrow \{z_3, z_5\}, d_6 \leftarrow \{z_1, z_4\}, d_7 \leftarrow \{z_1, z_2, z_5\}$. Three distinct topic-sets are hence present in $\mathcal{D}$, i.e., $\theta_1 = \{z_3, z_5\}, \theta_2 = \{z_1, z_4\}, \theta_3 = \{z_1, z_2, z_5\}$, which correspond to a 3-class partition of $\mathcal{D}$: $\{\{d_1, d_5\}, \{d_2, d_4, d_6\}, \{d_3, d_7\}\}$.

To compare clustering solutions and reference classification, we resorted to three widely used criteria in document clustering, namely *F-measure* (*F*), *Entropy* (*E*), and *Normalized Mutual Information* (*NMI*). Higher (resp. lower) values of *F* and *NMI* (resp. *E*) correspond to better clustering quality.

4.2 Results

We present here document clustering results obtained by SGM and the selected competing models, namely PLSA, LDA and Ext-PLSA. The generative processes of the various models were set in such a way that the topic variable assumed the same number of values as the number of topic-labels given for each dataset. Ext-PLSA also required a further latent variable related to the size of the desired clustering solutions. For this

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Table 2. SGM-based clustering performance on IEEE with different segmentations

segmentation setting	*#segments*	*F*	*E*	*NMI*
SGM^{avg}	155,828	0.64	0.58	0.49
SGM^{min}	89,539	0.59	0.62	0.45
SGM^{max}	179,491	0.58	0.60	0.47

Table 3. Summary of clustering results

	F				*E*				*NMI*			
	PLSA	Ext-PLSA	LDA	**SGM**	PLSA	Ext-PLSA	LDA	**SGM**	PLSA	Ext-PLSA	LDA	**SGM**
IEEE	0.53	0.56	0.46	0.64	0.70	0.73	0.62	0.58	0.37	0.32	0.44	0.49
PubMed	0.48	0.50	0.43	0.58	0.57	0.54	0.49	0.42	0.50	0.52	0.58	0.64
RCV1	0.49	0.54	0.42	0.56	0.57	0.59	0.51	0.48	0.49	0.46	0.54	0.59
avg score	*0.50*	*0.53*	*0.44*	*0.59*	*0.61*	*0.62*	*0.54*	*0.49*	*0.45*	*0.43*	*0.52*	*0.57*
avg gain	*+0.09*	*+0.06*	*+0.16*	—	*+0.12*	*+0.13*	*+0.05*	—	*+0.12*	*+0.14*	*+0.05*	—

study, we carried out the algorithms on CRESCO HPC system,[4] which is integrated into ENEA-GRID infrastructure. CRESCO is a general purpose system composed by 382 nodes with more than 3300 cores. We executed our experiments on a CentOS 5.5 platform, with Linux 2.6.18 kernel, 64GB memory, 4 Intel(R) Xeon(R) CPU E7330, 2.40GHz quadcore [16].

We initially investigated how clustering performance based on our SGM depends on the selected segmentation strategy. For this purpose, we tested SGM on the evaluation datasets by providing it with different input segmentations, namely SGM^{max}, SGM^{min}, and SGM^{avg}. We report results only for a selected dataset, as conclusions drawn from the remaining datasets were very similar to those here presented. Table 2 shows clustering results obtained on IEEE. In the table, we can observe that neither minimizing nor maximizing the number of segments improved the clustering accuracy obtained based on SGM^{avg}. Nevertheless, a higher number of segments would seem to be preferable to a smaller one. In fact, SGM^{max} achieved a little gain over SGM^{min} in terms of E and *NMI* (both around 0.02), while being comparable on F based evaluation. This can be explained since more segments would lead to discover (sub)topics that are hierarchically related to the main ones but also would tend to overfit the data, as the occurrences of any specific word will be diluted over the many segments and, consequently, such a topic-word over-specificity will correspond to more topic distributions.

Table 3 summarizes clustering results, where we used SGM^{avg}. A first evident remark is that our SGM led to the best clustering quality results. In fact, improvements in F-measure varied from 0.06 (vs. Ext-PLSA) to 0.16 (vs. LDA). Major improvements in terms of F-measure obtained by SGM were observed on IEEE and PubMed (above 0.08 on average better than the best among the competing models), whereas on RCV1 the performance gain was lower (about 0.02). This would indicate that benefits from text segmentation in document generative modeling are more evident for relatively long documents than short ones. Moreover, looking at the performance based on the other quality measures, average quality gains achieved by our SGM were quite similar to those previously discussed in terms of F-measure. In particular, SGM outperformed the other methods in terms of entropy, from 0.05 (vs. LDA) to 0.13 (vs. Ext-PLSA). In terms of *NMI*, quality improvements were from 0.05 (vs. LDA) to 0.14 (vs. Ext-PLSA). Comparing the performance of the competing methods, LDA outperformed both PLSA and

[4] http://www.cresco.enea.it/

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Table 4. Comparison with traditional VSM-based document clustering

	SGM-based clustering			VSM-based clustering		
dataset	*F*	*E*	*NMI*	*F*	*E*	*NMI*
IEEE	0.64	0.58	0.49	0.21	0.84	0.21
PubMed	0.58	0.42	0.64	0.31	0.79	0.28
RCV1	0.56	0.48	0.59	0.39	0.63	0.45
avg score	*0.61*	*0.49*	*0.57*	*0.30*	*0.75*	*0.31*

Ext-PLSA according to entropy and *NMI* (up to 0.11 *E* and 0.12 *NMI* both on IEEE), whereas Ext-PLSA behaved better than the other two methods in the case of F-measure evaluation (up to 0.12 *F* on RCV1). This would suggest that by using LDA clustering solutions tend to be less coarse than those obtained by PLSA and Ext-PLSA—because F-measure is typically biased towards coarser clustering.

Comparison with traditional document clustering. We also compared the performance achieved by clustering the segmented documents based on our SGM with an approach that first performs the clustering of the segments from the document collection (by treating each segment as a single mini-document), and finally derives a document clustering solution. For this purpose, in the baseline method segments were represented by the conventional vector-space model (VSM) equipped with the popular *tf.idf* term relevance weighting scheme. Clustering of the segments was performed by using the *Bisecting K-Means* [17] algorithm, which is widely known to produce high-quality, hard clustering solutions in high-dimensional, large datasets [18]. The segments that belong to the documents in the various collections were represented as conventional *tf.idf* vectors prior to inputting them to the clustering algorithm. Since the partitioning of the segment collection produced by Bisecting K-Means corresponds to a potentially soft clustering of the documents, we devised a simple method to derive a hard assignment of documents to clusters by adopting a majority voting strategy (i.e., each document is assigned to the cluster that contains the majority of its segments). Finally, the document clustering solution derived by this approach was evaluated w.r.t. the reference classification based on topic-sets for any specific dataset.

Table 4 summarizes results of this comparative analysis. SGM^{avg}-based clustering always outperformed the VSM-based clustering on all datasets, achieving average improvement per dataset of 0.31 *F*, 0.26 *E*, and 0.26 *NMI*. By modeling segmented documents, SGM was indeed able to directly produce a hard document clustering that corresponds to a finer mapping of documents to topic-sets, which is well-suited for better reflecting the multi-topic nature of documents. Conversely, by treating segments that belong to same document as independent text units to be clustered, the baseline document clustering approach tends to produce solutions whose document clusters are likely to be biased by those topics that are present in most of the segments within the same document.

5 Conclusions

We presented a generative model for topically-segmented documents, which introduces a segment model variable in the generative process. The topics of a document in a given collection are modeled as a mixture of the individual distributions of the topics present in each of the document segments. In this way, the bag-of-words assumption (which is typically exploited in statistical topic modeling) becomes more realistic since

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

it is transferred to smaller text units, i.e., the document segments. As experimentally proved, the topic modeling obtained on the within-document segments is better suited for documents that have an explicit multi-topic class structure.

References

1. Hofmann, T.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning **42**(1-2) (2001) 177–196
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research **3** (2003) 993–1022
3. Zhong, S., Ghosh, J.: Generative Model-Based Document Clustering: a Comparative Study. Knowledge and Information Systems **8**(3) (2005) 374–384
4. Sato, I., Nakagawa, H.: Knowledge Discovery of Multiple-Topic Document using Parametric Mixture Model with Dirichlet Prior. In: Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD). (2007) 590–598
5. Kim, Y.M., Pessiot, J.F., Amini, M.R., Gallinari, P.: An Extension of PLSA for Document Clustering. In: Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM). (2008) 1345–1346
6. Zeng, J., Cheung, W.K., Li, C., Liu, J.: Multirelational Topic Models. In: Proc. 9th IEEE Int. Conf. on Data Mining (ICDM). (2009) 1070–1075
7. Hearst, M.A.: TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. Computational Linguistics **23**(1) (1997) 33–64
8. Beeferman, D., Berger, A., Lafferty, J.: Statistical Models for Text Segmentation. Journal of Machine Learning Research **34**(1-3) (1999) 177–210
9. Choi, F.Y.Y., Wiemer-Hastings, P., Moore, J.: Latent Semantic Analysis for Text Segmentation. In: Proc. Int. Conf. on Empirical Methods in Natural Language Processing (EMNLP). (2001) 109–117
10. Brants, T., Chen, F., Tsochantaridis, I.: Topic-Based Document Segmentation with Probabilistic Latent Semantic Analysis. In: Proc. 11th ACM Int. Conf. on Information and Knowledge Management (CIKM). (2002) 211–218
11. Sun, Q., Li, R., Luo, D., Wu, X.: Text Segmentation with LDA-based Fisher Kernel. In: Proc. 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (HLT). (2008) 269–272
12. Shafiei, M.M., Milios, E.E.: A Statistical Model for Topic Segmentation and Clustering. In: Proc. Canadian Conf. on Artificial Intelligence. (2008) 283–295
13. Du, L., Buntine, W.L., Jin, H.: A segmented topic model based on the two-parameter Poisson-Dirichlet process. Machine Learning **81**(1) (2010) 5–19
14. Ponti, G., Tagarelli, A.: Topic-based Hard Clustering of Documents using Generative Models. In: Proc. 18th Int. Symp. on Methodologies for Intelligent Systems (ISMIS). (2009) 231–240
15. Kailath, T.: The Divergence and Bhattacharyya Distance Measures in Signal Selection. IEEE Transactions on Communication Technology **15**(1) (1967) 52–60
16. Bracco, G., Podda, S., Migliori, S., D'Angelo, P., Quintiliani, A., Giammattei, D., De Rosa, M., Pierattini, S., Furini, G., Guadagni, R., Simoni, F., Perozziello, A., De Gaetano, A., Pecoraro, S., Santoro, A., Scio', C., Rocchi, A., Funel, A., Raia, S., Aprea, G., Ferrara, U., Novi, D., Guarnieri, G.: CRESCO HPC System Integrated into ENEA-GRID Environment. In: Proc. of the Final Workshop of the Grid Projects of the Italian National Operational Program 2000-2006 – Call 1575. (2009) 151–155
17. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In: Proc. KDD'00 Workshop on Text Mining. (2000)
18. Zhao, Y., Karypis, G.: Empirical and Theoretical Comparison of Selected Criterion Functions for Document Clustering. Machine Learning **55**(3) (2004) 311–331

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Knowledge Discovery from Textual Sources by using Semantic Similarity*

Paolo Atzeni[1], Fabio Polticelli[2], and Daniele Toti[1]

[1] Department of Computer Science and Automation, Roma Tre University
atzeni@dia.uniroma3.it, toti@dia.uniroma3.it
[2] Department of Biology, Roma Tre University
polticel@uniroma3.it

Abstract. We propose a methodology to automatically discover characterizing knowledge from textual sources, with the purpose of semantically categorizing them and clustering them together according to their subjects. Such a methodology is based upon several challenging steps, like terminology extraction and disambiguation, semantic similarity identification via ontology alignment, and a core pattern-based strategy for automatic ontology building. This methodology was originally devised as an extension of PRAISED, our abbreviation identification and resolution proposal, with the purpose of allowing us to resolve previously unresolvable abbreviations, whose explanation either escapes the system's proximity-based approach or is not found within the very source text they are featured in. By moving from a paper-by-paper, mainly syntactical process to a corpus-based, semantic approach, it will be in fact possible to dramatically enhance our system in terms of its resolution capabilities. Nevertheless, the strategy we present here is not tied to this specific task, but is instead of relevance for a variety of contexts, and might therefore find a far wider applicability for other advanced knowledge extraction and discovery systems.

1 Introduction

Textual documents, either in their purely unstructured or semi-structured form, are undoubtedly the repository of most of the human knowledge. As the amount of available digital information grows to previously unimaginable levels, an unprecedented number of documents containing essential knowledge, albeit scattered among them, is at people's disposal for a variety of different studies and researches. All of the everyday manual operations involved for collecting and organizing such a knowledge are consequently getting more and more painstaking and time-consuming, thus yearning for semi-automatic or automatic solutions to help reduce costs in terms of both time and employed resources.

In this paper, we propose a methodology to automatically discover characterizing knowledge from scientific papers (full-texts), in order to sort them out according to the subjects they discuss, and therefore allow for speed-reading and

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

cataloging activities. This methodology is born as an extension of PRAISED [2–4, 13], our information extraction system whose main purpose is identifying and resolving abbreviations from full-text scientific papers. This abbreviation discovery process, as originally implemented, proceeded on a paper-by-paper basis, by deeming an abbreviation resolvable if and only if its explanation could be found within the same paper the former was featured in. Besides, a proximity-based scan was used to check for abbreviation explanations, thus potentially failing in resolving abbreviations whose explanation is mentioned in a whole difference section of the considered paper. By computing semantic similarity among papers from a given domain, we will show how it is possible to shift from a paper-by-paper to a corpus-based approach, so that abbreviations found in a determined text might be resolved by drawing on a different paper sharing similar subjects with the first. The strategy we discuss is made up of several substeps, involving tasks like terminology extraction, terminology disambiguation according to context, computation of semantic distance and automatic ontology building/learning and matching/alignment. It must be stressed out that such a strategy, despite being applied for the purpose of expanding the abbreviation discovery capabilities of our system, is relevant for a wider range of applications, and might be used as successfully to enhance other knowledge extraction systems as well.

The paper is structured as follows. In Section 2, related work is discussed. In Section 3, we briefly describe our abbreviation discovery system. In Section 4, we delve into the details of our strategy, by discussing its required steps and their expected outcome. And finally, in Section 5, we draw our conclusions.

2 Related Work

As far as abbreviation discovery is concerned, several research groups have proposed a certain number of methodologies, ranging from general approaches to more specific techniques. These include the use of regular expressions, linguistic cues and pattern-based recognition strategies, as well as machine learning algorithms, natural language processing and mixed methods. Even most of the major methods among them [10], [7], [1] possess several limitations (strong constraints, abstract-scope only, limited recall, no entity recognition etc.).

Regarding research areas like ontology building/learning and ontology matching/alignment, literature is vast and several potential approaches have been devised. A survey on ontology building methodologies is shown in [12]. Relevant proposals for ontology learning can be found in [14] and [5], although either they require human intervention or they show practical limitations even when dealing with simpler and shorter texts.

As far as ontology matching is concerned, [6] presents a thorough classification of ontology alignment methodologies and tools. One such tool is COMA [8], based on a decade-old experience in the field and largely used by the scientific community.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

3 The PRAISED System

PRAISED (currently defined as Processor for Abbreviation Identification, Disambiguation and Storage) was initially designed and implemented as an abbreviation discovery system for the biomedical community, focusing on the identification and resolution of protein abbreviations from full-text biological papers. Recently, it has been refined and generalized, by successfully applying it to other, non-biomedical domains. Further details can be found in [2–4, 13].

4 Knowledge Discovery Strategy

In this section we will provide the details for our knowledge discovery methodology, as composed by the three main steps further discussed below.

4.1 Step 1: Finding characterizing elements in a paper

The first step of the strategy takes as input a corpus of full-text papers, all sensibly belonging to a certain known domain, and produces as output an ontology for each of the papers making up the corpus: the ontology will represent the characterizing concepts detected from the text. This step, which involves several computationally-heavy tasks, is performed only once and one paper at a time, as a prerequisite for the subsequent steps, which are instead executed on demand.

Terminology extraction, disambiguation and classification In order to discover the characterizing elements of the considered paper, relevant terms must be extracted from the source text. This is a natural language processing task, where techniques like part-of-speech (POS) tagging, stemming and lemmatization play a central role. A sample text excerpt can be found in the leftmost part of Figure 1, taken from one of the Wikipedia entries for "Java", and will be used as an example throughout this discussion. We will refer to it as Paper 1.

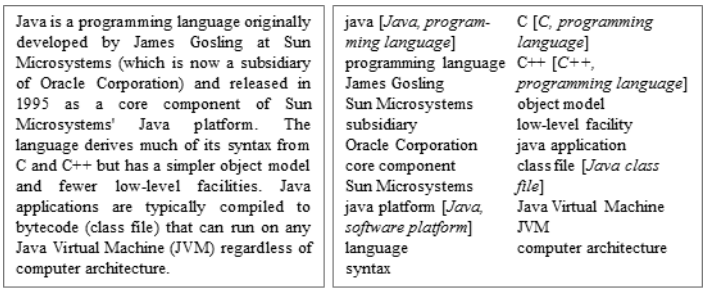


Fig. 1. A text excerpt (on the left) and its characterizing terms (on the right)

The operation pipeline is as follows:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

- **POS tagging**. The source text is tagged in order to identify the POS elements within it. For our examples, we use the Stanford NLP tools [11].
- **Candidate characterizing term selection and aggregation**. After the tagging is done, nouns, proper nouns and their combination with adjectives are selected as preliminary candidate characterizing terms. Adjacent nouns are combined as well to represent a single compound term.
- **Lemmatization**. Candidate terms from the previous selection are brought back to their lemma form, in order to ease the disambiguation to follow.
- **Terminology disambiguation and filtering**. Candidates thus identified are disambiguated according to words adjacent or next to them, so that polysemous terms are correctly associated with their actual meaning with respect of the lexical context they are placed in. This activity requires an external knowledge base: we take advantage of Wikipedia Miner for this purpose. The resulting candidate term list of Paper 1 after selection, aggregation, lemmatization and disambiguation is shown in the rightmost part of Figure 1.
- **Terminology classification**. Once the context information has been correctly associated with the candidate terms, they are given a score, or weight, based on the term frequency (TF); only those scoring higher than a set threshold will be actually selected. In the case of our sample text, due to its short length the majority of the detected terms share more or less the same score, for they appear only once in the text (with a few exceptions); the threshold setting is thus responsible to take into account the text's length, so that the classification procedure can work well with differently-sized input papers.
- **Abbreviation identification**. Along with the term list produced so far, a list of the abbreviations featured within the considered paper is also compiled and stored, by taking advantage of Phase 1 of PRAISED's discovery process. In Paper 1, only the term *JVM* is recognized as an abbreviation, and thus stored separately from the characterizing terms earlier produced (even though also featured among them).

Ontology building The next challenging task lies in tying together the characterizing terms extracted so far, by identifying relationships among them. This is what we mean with ontology learning or building, and comes down to tracing back, in an automatic fashion, explicit (or somewhat implicit) ties among terms.

In this regard, a pattern-matching strategy is employed to discover a certain number of interrelationships from the lexical contexts of the considered terms. Obviously, this kind of automatic detection could not claim completeness: only a selected number of relationships might be inferred in this fashion (e.g. is-a, equivalence, part-whole, property/relation etc.). On the other hand, by restricting the range of relationships to a similar subset, the semantic comparison to be performed in Step 2 can be smoothed and produce more effective results.

Let us review this process by considering our example. In Figure 2 we see the resulting ontology automatically built from the source text categorized in Step

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

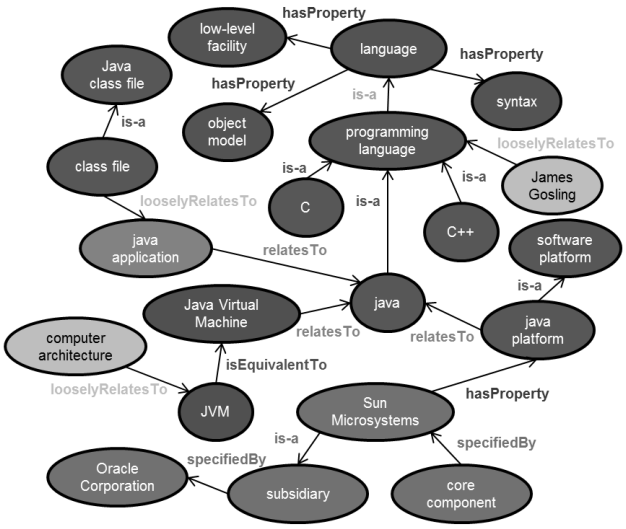


Fig. 2. Ontology automatically built from the characterizing terms obtained in Step 1 for Paper 1

1. Relationships correlating the characterizing terms of the input text are built in the following order and with the following strategies (each step corresponding to a different relationship color in the figure; the background color of an element underlines the substep in which that element first appears in the ontology):

- is-a relationships derived from terminology disambiguation (blue color);
- equivalence relationships, via equivalence patterns (following parenthesis-enclosed explanation, correlation expressions like "as known as" etc.) (purple color);
- is-a relationships from lexical patterns (verb "to be" + article, relative connectors + verb "to be", "such as" etc.), and specification relationships (from "of" connectors) (green color);
- part-of relationships from expressions like "is made of" etc., and property-owning relationships from verb "to have", possessive adjectives and similar expressions (brown color);
- is-a relationships derived from lexical inclusion of characterizing terms (as in "programming language", which is a "language") (olive color);
- general relationships from terms sharing an adjective/a specifying element to the actual term shared (as in "Java Virtual Machine", "java platform", "java application", which are all related to "java") (light blue color);
- loose relationships for tying terms either isolated or yet to be correlated, according to proximity and appearance in the same sentence (the uncorrelated "computer architecture" to "class file", or the isolated "class file" to "java application") (yellow color).

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Some imprecisions can be noticed in the automatic building. For instance, "class file", earlier disambiguated as "java class file", ends up being a subclass of "java class file", whereas it should be the other way around; also, a significant term like James Gosling, the Java creator, gets loosely tied to "programming language" instead of "java", due to the inability of inferring a specific relationship between it and the term "java". There are of course margins of improvement for the automatic ontology building process based on lexical patterns.

In the end, though, the results of Step 1 will be as many ontologies as the papers processed, along with an abbreviation list for each of them. This way, a full-text corpus can be automatically categorized with semantic information for the papers it includes.

4.2 Step 2: Computing semantic similarity

Once the paper corpus has been properly semantically categorized, it is possible to proceed with the on-demand steps. The second step takes place whenever a paper, scanned by PRAISED for abbreviations, ends up featuring an abbreviation without a corresponding explanation. This may happen for two reasons: either the abbreviation explanation escapes the proximity-based approach implemented by PRAISED (ending up in a different sentence or a different section of the document altogether), or it is simply not present within that very paper. In order to try and resolve such an unresolved abbreviation nevertheless, we need to identify those papers bearing the highest similarity with the original text in terms of the subjects discussed. For this purpose, the ontologies created in Step 1 must be purposefully compared.

A Java Virtual Machine is a piece of software that is implemented on non-virtual hardware and on standard operating systems. A JVM provides an environment in which Java bytecode can be executed, enabling such features as automated exception handling, which provides "root-cause" debugging information for every software error (exception).	Java Virtual Machine piece software [*Computer software*] non-virtual hardware standard operating system JVM environment java bytecode feature automated exception handling root-cause debugging information software error [*Software bug*] exception

Fig. 3. A text excerpt featuring an unresolvable abbreviation (on the left) and its characterizing terms (on the right)

Let us consider the text in Figure 3, taken from the Wikipedia entry for "Java Virtual Machine", which we will refer to as Paper 2, along with its characterizing terms. Such an excerpt features an abbreviation, *JVM*, which escapes PRAISED's proximity approach for its abbreviation resolution phase: thus, it cannot be resolved by the system. This text turns into the ontology in Figure 4 after Step 1. Incidentally, such an ontology features an island of terms isolated from the rest of the structure: this might not be allowed in certain formalizations.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

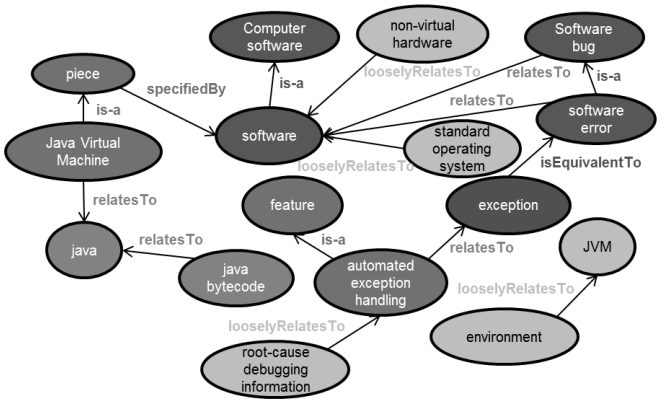


Fig. 4. Resulting ontology for Paper 2

Ontology alignment The process of comparing ontologies with each other is usually defined as ontology matching or alignment. Given two ontologies, it is paramount to try and identify terms which are both alike and related to all or some of the same concepts as well. For this purpose, we first rely onto the lexical level, by checking the mutual distance among the terms of the source and target ontology, via the Jaccard and weighted edit distance; afterwards, we focus on the ontological structure, and compute a comparison in terms of the kinds of relationships defined between elements from the considered ontologies. The result of this matching returns a similarity score: those papers deemed most similar will be the candidates where unresolvable abbreviations from the original paper might be indeed found. During this phase, the abbreviation lists from Step 1 is also taken into consideration: the amount of abbreviations shared between papers affects the final similarity score, which is increased accordingly. In the example proposed, Paper 2 is compared to the other papers in the corpus in terms of their ontologies. The ontology built from Paper 2 and the one built from Paper 1 are eventually assigned a high similarity score, since they share several similarities at various levels.

4.3 Step 3: Tracing back abbreviation explanations

The final step is simply a matter of applying the PRAISED process to the papers most similar to the one considered. From our example, Paper 1, once assigned a high similarity score with respect of Paper 2, will be a fit candidate to be used as the search space for the unresolved abbreviation (*JVM*) found in Paper 2. Indeed, the explanation of such an abbreviation is actually featured in Paper 1 (and appears in its abbreviation list as well), and will be successfully matched with its corresponding abbreviation by PRAISED's resolution process, this way resolving the original unresolved abbreviation from Paper 2.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

5 Conclusions

In this paper we have proposed a methodology to discover characterizing knowledge from a corpus of full-text papers, in order to assess their semantic proximity and be able to categorize them according to similar topics. This strategy widens the scope of our abbreviation discovery system to a corpus-based approach, where an abbreviation explanation may be detected in any other, semantically similar text from a given corpus. At the same time, this methodology might in principle be applied to different contexts as well, for no constraints tie it to the specific task at all. At the present time, we are finalizing its implementation: some candidate tools have already been sorted out, integrated and tested for several of its substeps, whereas we are currently fine-tuning and experimenting our core ontology building algorithm. Minor refinements notwithstanding, we believe that such a methodology will end up representing a step forward both in ours and in other knowledge extraction systems.

References

1. H. Ao and T. Takagi. Alice: an algorithm to extract abbreviations from medline. In *J. Am. Med. Inform. Assoc.*, 12, pages 576-586, 2005.
2. P. Atzeni, F. Polticelli and D. Toti. A Framework for Semi-Automatic Identification, Disambiguation and Storage of Protein-related Abbreviations in Scientific Literature. In *ICDE, DaLi Workshop*, 2011.
3. P. Atzeni, F. Polticelli and D. Toti. An Automatic Identification and Resolution System for Protein-related Abbreviations in Scientific Papers. In *EvoBio*, 2011.
4. P. Atzeni, F. Polticelli and D. Toti. Experimentation of an Automatic Resolution Method for Protein Abbreviations in Full-Text Papers. In *ACM-BCB*, 2011.
5. P. Cimiano and J. Vlker. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. In *NLDB*, 2005.
6. J. Euzenat et al. State of the art on ontology alignment. In *Knowledgeweb*, 2004.
7. C. Kuo, M. HT Ling, K. T. Lin and C. N. Hsu. BIOADI: a machine learning approach to identifying abbreviations and definitions in biological literature. In *BMC Bioinformatics*, Vol. 10, 2009.
8. S. Massmann, S. Raunich, D. Aumueller, P. Arnold and E. Rahm. Evolution of the COMA Match System. In *OM-2011*, 2011.
9. D. Milne and I.H. Witten. An open-source toolkit for mining Wikipedia by: D. Milne, and I.H. Witten. In *NZCSRSC*, Vol. 9, 2009.
10. A. Schwartz and M. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical texts. In *PSB*, 2003.
11. The Stanford NLP Group. http://nlp.stanford.edu/index.shtml
12. R. Subhashini and J. Akilandeswari. A survey on ontology construction methodologies. In *IJECBS (Online)*, Vol. 1, No. 1, 2011.
13. D. Toti, P. Atzeni and F. Polticelli. Automatic Protein Abbreviations Discovery and Resolution from Full-Text Scientific Papers: The PRAISED Framework. In *Bio-Algorithms and Med-Systems (BAMS)*, Vol. 8, No. 1, 2012.
14. Y. Wang, J. Vlker and P. Haase. Towards Semi-automatic Ontology Building Supported by Large-scale Knowledge Acquisition. In *AAAI Fall Symposium On Semantic Web for Collaborative Knowledge Acquisition*, Vol. FS-06-06, pages 70-77, 2006.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Mining Temporal Evolution of Criminal Behaviors

Gianvito Pio, Michelangelo Ceci, and Donato Malerba

University of Bari "Aldo Moro"
Department of Computer Science - Via Orabona, 4 - 70125 Bari, Italy
gianvito.pio@uniba.it, ceci,malerba@di.uniba.it

Abstract. One of the recently addressed research directions focuses on the issues raised by the diffusion of highly dynamic on-line information, particularly on the problem of mining topic evolutions from news. Among several applications, risk identification and analysis may exploit mining topic evolution from news in order to support law enforcement officers in risk and threat assessment. Assimilating the concept of *topic* to the concept of *crime typology* represented by a group of "similar" criminals, it is possible to apply topic evolution mining techniques to discover evolutions of criminal behaviors over time. At this aim, we incrementally analyze streams of publicly available news about criminals (e.g. daily police reports, public court records, legal instruments) in order to identify clusters of similar criminals and represent their evolution over time. Experimental results on both real world and synthetically generated datasets prove the effectiveness of the proposed approach.

1 Introduction

Over the last years, Topic Detection and Tracking (TDT) [3, 25, 5] is being recognized as an important research area in data mining. According to the classification suggested in [10], there are three main research lines in TDT:

- *News segmentation:* news coming from streams are clustered according to their topic. Each cluster represents the same topic across the time dimension.
- *New topic detection*: new clusters are identified from streams of news.
- *Topic tracking*: evolutions of clusters are tracked. In this case, incoming news can be associated to existing clusters, causing changes in clusters' properties.

By focusing our attention on topic tracking, in this paper, we argue that it is possible such techniques to discover evolutions of criminal behaviors over time. This approach would lead to gain useful knowledge that law enforcement officers can profitably exploit in risk and threat assessment. In general, studies in the literature have proved the effectiveness of data mining techniques in supporting the investigative activity [9, 15, 22] in risk identification and analysis. In this respect, we contribute by proposing a method that incrementally analyzes streams of news in order to identify clusters of "similar" criminals and represent their evolution over time. At this purpose, we apply a time-slice density estimation

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

method [2] that allows us to represent and analyze the evolution of the profile associated to each criminal by measuring the rate of change of criminal activities' properties and peculiarities over a time horizon. The proposed approach is tailored to deal with news, that are relatively easy to collect but are, in general, short and tend to spread and fade in volume on a specific topic in relatively small time intervals. This means that it is necessary to model the evolutions of underlying activities/phenomena by analyzing small textual documents.

This paper studies the problem of mining evolutions in terms of understanding the nature of the data changes in streams of news. Indeed, in the literature, several papers have faced this task and, in particular, the problem of tracking topics, ideas and "memes" from news [17, 27]. However, differently from papers found in the literature, in this work we do not consider variations and evolution (in volume) of short and distinctive phrases in the news, but the evolution (about properties associated to crime activities) of each single criminal to which multiple news can be associated. An evolution is expressed according to the relevant terms that allow us the representation and characterization of criminals. From the methodological viewpoint, we do not identify the evolution by evaluating whether a particular data mining model has become stale or not because of the underlying change in the data distribution [16, 2], but we provide the user with an understanding of the changes by eliciting the relevant terms and their weight.

The contribution of this paper is manifold: *i)* to define an unsupervised feature selection algorithm that is tailored for representing news; *ii)* to represent the criminal's profile and to on-line modify it according to more recent news; *iii)* to generate clusters of similar criminals and represent and analyze their evolution.

The paper is organized as follows: in the next section, related work on the considered application domain and methodological aspects are discussed; Section 3 is devoted to the detailed presentation of our method; in Section 4 the empirical evaluation on both artificially generated and real-world datasets is reported. Finally, conclusions are drawn and some future work is outlined.

2 Related Work

In the literature, a variety of approaches to deal with evolving clusters from textual data streams can be found. For example, in [26] the authors propose an incremental and neural network based version of the spherical k-means [11] which, according to an appropriate rule for updating the weights of the neural network, incrementally modifies the closest cluster center, given a new incoming news. A different approach is proposed in [1], where the authors face the problem of clustering blogs by considering labels manually associated to them and generating, in this way, a so called "collective wisdom". In [18] the authors propose a system called Personalized Blog Reader (PBR), which clusters stories into topics from different blogs. Clustering is performed in two phases: a static clustering algorithm, which provides a set of initial clusters, and a dynamic clustering algorithm, which incrementally updates clusters on the basis of the distance between new stories and clusters' stories. Despite the clear relationship,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

there are important differences between topic tracking and the research we report in this paper. First, in topic tracking clusters represent the same topic across the time dimension. In our case, each cluster is associated to a single time interval. Consequently, we do not apply classical incremental clustering approaches, but we identify clusters for each time interval and compare them with clusters identified for previous time intervals. Second, in topic tracking, clusters group together news about the same topic (i.e. the unit of analysis is the news), while in our case the considered unit of analysis is the person the news is associated with (the criminal). This means that we cluster together similar criminals on the basis of news associated to them.

A similar approach to ours is proposed in [4], where clusters of keywords extracted from messages published in blogs are identified for each time interval. Clusters associated to consecutive time intervals are pairwise compared in order to identify pairs with the highest affinity. By combining affinity relationships over several time intervals, it is possible to identify the top-k paths that express the most significant evolutions of the initial clusters over time. The main difference with respect to our approach is that the considered unit of analysis is the "keyword" (and not, for example, the author of the blog), on the basis of the intuition that clusters of keywords characterize topics.

Another approach, which is similar to ours, is presented in [23], where the authors propose an approach for defining and monitoring blog interests by clustering, for each time interval, blogs on the basis of their content. However, in this case, clustering is performed on the pairs $(class, similarity)$ obtained by a centroid based classifier. This means that clustering significantly depends on a preliminary supervised learning phase.

As for the specific application domain, in the seminal work in [9], the authors propose the use of different data mining algorithms (for clustering, association rule mining sequential pattern mining, deviation detection, classification and social network analysis) for analyzing data about criminal activities, in order to solve specific problems such as identification of possible money laundering, characterization of criminals' profiles, network intrusion detection, etc. Following this seminal work, most of the research has mainly focused on (social/criminal) network link analysis [21, 20] and crime entity associations extraction from textual documents [9, 8, 24]. However, at the best of our knowledge, no work considers the problem of mining evolutions of criminal behaviors over time.

3 TB-CREDIS

In this section we define the method *TB-CREDIS (Time-Based CRiminal Evolution DIScoverer)* that allows us to discover *criminal evolutions* from textual documents which are related to crimes. In this respect, a *criminal evolution* can be defined as a relevant change of crime typologies committed by a group of similar criminals in different time windows. All the necessary information are extracted from time-stamped textual documents (the time-stamp can represent the publication date) which are implicitly associated to a single time window.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

In this work, *time windows* are defined as adjacent and disjunct time intervals, obtained by subdividing the entire period we intend to analyze into intervals of the same size. Evolutions are discovered by analyzing the changes identified among distinct time windows.

The textual content of each document d_j is represented according to the classical *Vector Space Model (VSM)*, i.e. as an m-dimensional vector, where m is the number of extracted terms after the application of classical pre-processing methods for tokenization, stop-word removal and stemming. Each document vector contains normalized *TF-IDF* values associated to term-document pairs.

In the same way, each criminal is represented as a vector, in the same terms space used for documents (an example that we will show in the experiments is: [attack; fire; claim; suspect; report; injur; islam]), which better represents his *semantic position* and describes his crime typologies in a given time window. Since the terms space can be very large, a feature selection phase is necessary.

Summarizing, the method *TB-CREDIS* consists on the application of the following phases: *i) VSM* representation of the documents, *ii)* feature selection, *iii)* identification of the semantic position of each criminal in each time window, *iv)* clustering of criminals for each time window and *v)* evolution discovery and analysis.

In the following subsections, we explain the methods we use for performing feature selection, representing criminals and studying their evolution.

3.1 Feature Selection

In most of the feature selection methods proposed in the literature, the goal is to analyze the correlation between each attribute (or a set of attributes) and the target attribute, in order to favor attributes with the highest prediction power. However, in our case, we do not have any target attribute to guide the selection, that has to be performed in an unsupervised way (see [6] for related works on unsupervised feature selection algorithms). In this paper we use two distinct unsupervised feature selection algorithms. The first one is used as a baseline criterion and is based on the variance of the feature values, while the second (MIGRAL-CP) rewards features that show both great variation for dissimilar examples and low variation for similar ones and, at the same time, penalizes features that are correlated to already selected ones (penalizes redundant features).

Variance The most straightforward way to select a subset of terms is by computing the variance of the relative term-frequency of each term in the entire document collection and keeping the k terms with the highest variance. Intuitively, a term with a high variance in its frequency will better discriminate documents, whereas a term with low or zero variance in its frequency will substantially describe the documents in the same way.

This feature selection algorithm has the advantage of a linear time complexity, at the price of some disadvantages:

– it selects the terms which best discriminate between documents, disregarding their real similarity. In fact, it does not take into account the case in which

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

similar documents share the same terms with similar relative term-frequency. This can lead to lose terms that characterize entire classes of documents.
– it does not consider the correlation between terms. In fact, two strongly correlated terms will be both selected if they are in the set of the top k terms with the highest variance. This leads to select redundant terms.

MIGRAL-CP A more sophisticated attempt to perform unsupervised feature selection is reported in [13], where the authors propose to use the Laplacian Score to identify the features which better preserve samples similarity. However, the Laplacian Score rewards features for which similar samples show a small variation in the feature values, but does not reward those that show a large variation for dissimilar samples. Starting from this work, we define a different method called *MInimum GRAph Loss with Correlation Penalty (MIGRAL-CP)*. The goal of this method is twofold: *i)* to select features (terms) which show both great variation for dissimilar examples (documents) and low variation for similar ones and *ii)* to discard features correlated with already selected features. Both goals are of fundamental importance for the task at hand.

Formally, given the set of documents $D = \{d_1, d_2, \ldots, d_n\}$ and the complete set of terms $T = \{t_1, t_2, \ldots, t_m\}$, we build the (fully-connected undirected) neighborhood graph G, where each node represents a document and each edge is labeled with the similarity value between the corresponding documents. The label of the edge between the nodes associated to the documents d_i and d_j is defined as: $v_{ij} = e^{-\left|\left|w_{d_i} - w_{d_j}\right|\right|^2}$, where w_{d_i} (w_{d_j}) is the relative term-frequency vector associated to the document d_i (d_j), defined according to the entire set of terms T. The similarity measure directly follows from the Laplacian score definition but, obviously, any other similarity measure might be considered.

We define an iterative method to select a subset of k terms which satisfy the above requirements. The first term is selected in order to maximize the score:

$$Score_1(t_r) = \frac{1}{2}\left(1 - \frac{1}{n}\sum_{j=1}^{n}\rho(V_j, F_{r,j})\right) \tag{1}$$

where:

– $V_j = [v_{j,1}, v_{j,2}, \ldots, v_{j,n}]$ are the similarity values between the document j and all the other documents, using all the terms.
– $F_{r,j} = \left[(s_{r,j} - s_{r,1})^2, (s_{r,j} - s_{r,2})^2, \ldots, (s_{r,j} - s_{r,n})^2\right]$ are the dissimilarities between the document j and all the other documents, using the term t_r only. In this formula, $s_{r,j}$ is the relative term frequency of the term t_r in d_j.
– $\rho(\cdot, \cdot)$ is the classical Pearson correlation coefficient.

The first selected term ($\hat{t}_1 = \arg\max_{t_r \in T} Score_1(t_r)$) will be the one which determines the highest inverse linear correlation between the documents' similarities computed with all the terms and documents' dissimilarities computed with only that term. The remaining $k-1$ terms are selected according to:

$$Score_i(t_r) = Score_{i-1}(t_r) \times (1 - penalty(t_r, \hat{t}_{i-1})) \tag{2}$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

which, at each iteration i, reduces the score associated to each term according to a penalty function $penalty(t_r, \hat{t}_{i-1})$ which takes into account the correlation between a term and the term that has been selected as feature in the previous iteration. In this way, we prevent that two redundant terms are both considered as features. Coherently with the correlation coefficient introduced before, here we introduce the following definition: $penalty(t_r, \hat{t}_{i-1}) = \max(0, |\rho(t_r, \hat{t}_{i-1})| - \gamma)$, where $0 \leq \gamma \leq 1$. The rationale of this choice is that a correlation value of $|\rho(t_r, \hat{t}_{i-1})| \leq \gamma$ is considered too small to result in a penalty.

3.2 Representing Criminals

For each time window τ_z, the semantic position of each criminal is represented in the same k-dimensional terms space identified in the feature selection phase. In the following we describe two possible alternatives.

Time-Weighted Centroid The most straightforward approach is to compute a weighted centroid of the vectors of the documents associated to the criminal, such that a higher weight is assigned to more recent documents. Formally, the semantic position of the criminal c for the time window τ_z is computed as:

$$X(c, \tau_z, h) = \frac{\sum_{<d_j, \tau_j> \in S_{c,\tau_z,h}} p_{\tau_z,\tau_j}(h) \times w_{d_j}}{\sum_{<d_j, \tau_j> \in S_{c,\tau_z,h}} p_{\tau_z,\tau_j}(h)}, \tag{3}$$

where $S_{c,\tau_z,h}$ is the set of documents associated to the criminal c, belonging to the considered time window τ_z or one of the previous $h-1$ time windows, and $p_{\tau_z,\tau_j}(h) = 1 - \frac{z-j+1}{h}$ is the time fading-factor which reduces the effect of the document d_j according to the distance between the considered time window (τ_z) and the time window τ_j the considered document is associated to.

Max Density Point A more sophisticated solution for representing criminals comes from the work in [7]. This solution consists in substituting each point (document) with a k-dimensional Gaussian function and searching for the point in the space with the highest sum of contributions. Formally, each document d_j is represented as a function $d'_j(\cdot)$ with domain in $[0,1]^k$ and range in $\mathbb{R}^+$:

$$d'_j(x) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - s_{i,j})^2}{2\sigma^2}} \tag{4}$$

where $\sigma \in [0,1]$ is a parameter that defines the width of the Gaussian function.

The position of the criminal c for the time window τ_z is computed as follows:

$$X(c, \tau_z, h) = \arg\max_{x \in [0,1]^k} \sum_{<d_j, \tau_j> \in S_{c,\tau_z,h}} p_{\tau_z,\tau_j}(h) \times d'_j(x) \tag{5}$$

where the time fading-factor $p_{\tau_z,\tau_j}(h)$ reduces the value of the Gaussian function.

Since it is computationally impractical to search for the point $X(c, \tau_z, h)$ in the continuous space $[0,1]^k$, an equal-width discrete space is used, instead. This

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

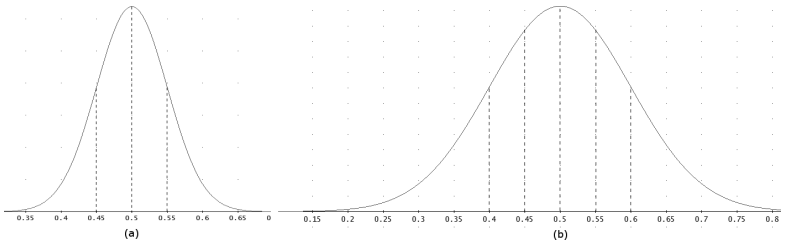


Fig. 1. Gaussian function defined on a single dimension with $y = 0.5$, $\beta = 20$, $\sigma = 0.05$ (a) and $\sigma = 0.10$ (b). In (a) it would be enough to analyze only the values 0.45, 0.50 and 0.55, whereas in (b) it would be necessary to analyze also the values 0.40 and 0.60.

discrete space is defined as Φ^k, where $\Phi = \left\{0, \frac{1}{\beta}, \frac{2}{\beta}, \ldots, \frac{\beta-1}{\beta}, 1\right\}$ and $\beta+1$ is the number of desired distinct values. The value of each dimension is associated to its nearest value in Φ through a function $\psi : [0,1] \rightarrow \Phi$.

Although we work in the discrete space, the computational complexity remains high, that is, exponentially depending on the number of selected terms k and linearly depending on the number of documents associated to each criminal $|S_{c,\tau_z,h}|$. However, the computation can be further optimized according to two criteria. *i)* The search can be limited to the areas interested by at least one document. *ii)* We can adopt a greedy search that focuses only around the points for which the $d'_j(\cdot)$ functions[1], which contribute to $X(c,\tau_z,h)$, reach the maximum values. For instance, consider the case depicted in Figure 1, it is possible to focus on the search in a smaller area on the left and on the right (for each dimension) of the points for which the bell-shaped functions assumes the maximum value.

Let y be the value assumed on a given dimension by a given document. Given the applied discretization, we analyze only $\beta\sigma$ values on both sides of y, leading to a total of $2\beta\sigma + 1$ values for each dimension[2], instead of $\beta + 1$. In this way, we cover all the available values in $[y - \sigma; y + \sigma]$.

Furthermore, an *incremental* approach can be applied, in order to take advantage of the result obtained for the previous time window. In fact, the position $X(c,\tau_z,h)$ can be either the same of the previous time window ($X(c,\tau_{z-1},h)$) or a new position around a new document only. Thus the search can be limited to the areas interested by only the documents belonging to τ_z and to $X(c,\tau_z,h)$.

Let $\Phi^k_{S_{c,\tau_z}} \subseteq \Phi^k$ the set of values obtained after the application of the optimization criteria *i)* and *ii)*, the semantic position is calculated as follows:

$$X(c,\tau_z,h) = \underset{x \in (\Phi^k_{S_{c,\tau_z}} \cup \{X(c,\tau_{z-1},h)\})}{\arg\max} \sum_{<d_j,\tau_j> \in S_{c,\tau_z,h}} p_{\tau_z,\tau_j}(h) \times d'_j(x). \quad (6)$$

Finally, the criminal position $X(c,\tau_z,h)$ in (6) can be calculated in a parallel way, assigning at each computational unit (i.e., available CPU) a subset of $\Phi^k_{S_{c,\tau_z}} \cup \{X(c,\tau_{z-1},h)\}$ and aggregating results.

[1] $d'_j(\cdot)$ is a bell-shaped function in the k-dimensional space.

[2] Reasonable values of σ are ≤ 0.1. In the experiments we use $\sigma \in \{0.05, 0.1\}$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

3.3 Clusters evolution discovering

The last necessary step, before analyzing criminal evolutions, consists of searching for homogeneous groups (i.e. clusters) of criminals for each time window. At this purpose, we use a modified version of the K-means algorithm. This choice is motivated by the fact that it is a well known standard algorithm that would provide a fair comparison between the feature selection algorithms and between the criminal representation criteria. Obviously, in our framework, any clustering algorithm (also density based, such as DBSCAN [12]) can be plugged in.

Our improvement to the standard K-means algorithm consists in the automatic identification of the reasonable number of clusters to be extracted. Indeed, this is necessary in the task at hand, where the evolution of criminal behaviors does not allow the user to a-priori define the number of clusters for each time window. Moreover, there is no guarantee that all the crime typologies are necessarily present in each time window. The solution we adopt to solve this problem is that of exploiting *Principal Component Analysis (PCA)*. In general, the idea behind PCA is that of identifying principal components (new orthogonal axes) according to which data are distributed. That is, given a set of feature-vector represented examples, PCA identifies a new (smaller) set of features, which are obtained as a linear combination of the original ones. By inverting the roles of features and examples, it is possible to identify new (orthogonal) examples (prototype vectors), according to which other examples (vectors) can be aggregated. Since PCA identifies the smaller number of prototype vectors such that a given percentage of the variance in the data is explained [14], the number of prototype vectors can be used as an indication of the appropriate number of clusters.

Once clustering is performed for each time window, it is possible to identify:

- the *position* of the cluster in the k-dimensional terms space. Each cluster is usually identified by a number, which gives no information about the cluster semantic. Analyzing the terms with the highest weights in the cluster (e.g. of its centroid) can give an idea about the crime typology it represents.
- a *matching* between clusters of different time windows according to the similarity between the clusters' centroids (Figure 2). Unfortunately, there is no guarantee that it is always possible to find meaningful matches.
- the *number of criminals* which have evolved from the crime typology represented by C_i to that represented by C_j, where C_i and C_j are two generic clusters extracted for the time windows τ_{z-1} and τ_z, respectively (Figure 3).

4 Empirical Evaluation

In this section we evaluate the proposed method on both synthetic and real data.

As regards the synthetic dataset, news about 100 criminals are generated for ten consecutive annual time windows (from 2001 to 2010). For each criminal, a set of less then 200 news, each belonging to one of the considered time windows, is generated by considering 7 specific vocabularies and a generic English vocabulary, in order to introduce "noise terms". For each time window and criminal,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

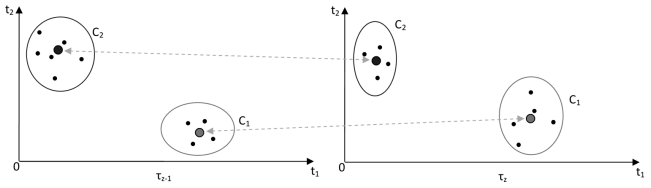


Fig. 2. An example of matching found between two clusters belonging to different time windows, analyzing the centroids' similarity.

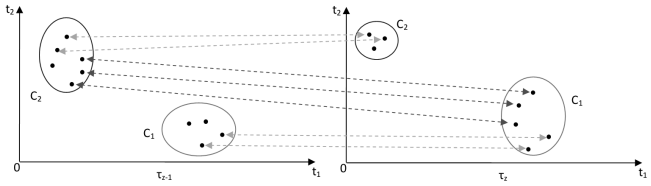


Fig. 3. Example of a discovered criminal evolution. Three criminals have moved from the cluster C_2 in τ_{z-1} to the cluster C_1 in τ_z.

the probability of changing crime typology is set to 0.3. The feature selection phase is executed with both the proposed methods, with $k = 10$ and $\gamma = 0.5$ (for MIGRAL-CP). For MaxDensity, β is set to 20. We remind that k is the number of features to select, γ is the minimum value of the Pearson coefficient (between the last selected feature and the other features) to apply a penalty and $\beta + 1$ is the number of distinct values in which to discretize the interval [0,1].

As regards the real dataset, we consider the Global Terrorism Database (GTD)[3]. GTD consists of information on terrorist events (currently more than 98000 cases) around the world from 1970 through 2010 (with annual updates), including systematic data (news, criminals and timestamp) on domestic as well as international terrorist incidents. In this experiment, we consider 13 consecutive annual time windows (from 1998 to 2010), for a total of 11225 news concerning 82 criminals (or criminal organizations). The feature selection phase is executed with both the proposed methods, with $k = 15$ and $\gamma = .5$ (for MIGRAL-CP). For MaxDensity, β is set to 20.

Results are collected in terms of a variant of the Q-Modularity measure [19], which allows us to evaluate the quality of the clustering with respect to a random clustering. Let $e_{ij} = \frac{2}{r(r-1)} \sum_{c' \in C_i, c'' \in C_j} sim(X(c', \tau_z, h), X(c'', \tau_z, h))$ be a measure of the strength of the interconnections between criminals in the cluster C_i and criminals in the cluster C_j. In this formula, r represents the total number of criminals and $sim(\cdot, \cdot) \in [0, 1]$ is the cosine similarity. Intuitively, we want clusters for which e_{ii} values are generally large and $e_{ij} (i \neq j)$ values are generally small. Formally: $Q = \sum_{i=1}^{k} \left(e_{ii} - a_i^2\right)$, where $a_i = \sum_j e_{ij} = \sum_j e_{ji}$.

[3] `http://www.start.umd.edu/gtd/`

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Table 1. Clustering Q-Modularity and running times on the synthetic dataset.

				Variance		MIGRAL-CP	
Position	**h**	σ	**Var**	**time**	**q-mod**	**time**	**q-mod**
Centroid	2	-	80%	00:20:52	0.157	00:55:54	0.198
Centroid	2	-	90%	00:20:52	0.150	00:55:54	0.209
Centroid	2	-	80%	00:20:58	0.102	00:55:59	0.142
Centroid	2	-	90%	00:20:58	0.101	00:55:59	0.143
Centroid	2	-	80%	00:21:00	0.080	00:56:01	0.114
Centroid	2	-	90%	00:21:00	0.081	00:56:01	0.115
MaxDensity	2	0.05	80%	00:21:47	0.322	00:56:44	0.392
MaxDensity	2	0.05	90%	00:21:47	0.356	00:56:44	0.380
MaxDensity	2	0.10	80%	01:20:35	0.335	01:50:08	0.375
MaxDensity	2	0.10	90%	01:20:35	0.357	01:50:08	0.373
MaxDensity	5	0.05	80%	00:20:19	0.341	00:57:12	0.399
MaxDensity	5	0.05	90%	00:20:19	0.365	00:57:12	0.379
MaxDensity	5	0.10	80%	01:53:13	0.363	02:19:22	0.350
MaxDensity	5	0.10	90%	01:53:13	0.366	02:19:22	0.368
MaxDensity	10	0.05	80%	00:22:27	0.339	00:57:28	0.386
MaxDensity	10	0.05	90%	00:22:27	0.385	00:57:28	0.371
MaxDensity	10	0.10	80%	02:10:17	0.369	02:35:54	0.354
MaxDensity	10	0.10	90%	02:10:17	0.372	02:35:54	0.369

Table 2. Clustering Q-Modularity and running times on the GTD dataset.

				Variance		MIGRAL-CP	
Position	**h**	σ	**Var**	**time**	**q-mod**	**time**	**q-mod**
Centroid	2	-	90%	00:09:01	0.294	39:54:44	0.245
Centroid	2	-	95%	00:09:01	0.319	39:54:44	0.270
Centroid	5	-	90%	00:09:06	0.297	39:54:48	0.224
Centroid	5	-	95%	00:09:06	0.316	39:54:48	0.249
Centroid	10	-	90%	00:09:10	0.304	39:54:51	0.232
Centroid	10	-	95%	00:09:10	0.322	39:54:51	0.245
MaxDensity	2	0.05	90%	110:41:36	0.322	100:17:46	0.447
MaxDensity	2	0.05	95%	110:41:36	0.509	100:17:46	0.479
MaxDensity	5	0.05	90%	137:41:36	0.325	118:59:17	0.454
MaxDensity	5	0.05	95%	137:41:36	0.521	118:59:17	0.487
MaxDensity	10	0.05	90%	144:20:21	0.400	126:06:27	0.452
MaxDensity	10	0.05	95%	144:20:21	0.524	126:06:27	0.479

As it can be observed in Tables 1 and 2, the MIGRAL-CP algorithm leads to higher clustering quality in the synthetic dataset, with respect to the variance-based method, at the price of significantly higher running times. This conclusion does not hold for the GTD dataset, where there is no clear advantage of MIGRAL-CP over the variance-based method in the case of MaxDensity (where we have better results). As regards the method for computing the criminals' position, the MaxDensity method always significantly outperforms the centroid method on both datasets, at the price of a slightly higher running times. A more detailed analysis reveals that the best combination is MaxDensity-Variance in the case of a relatively small number of clusters (Var=90%) and MaxDensity-MIGRAL-CP in the case of a relatively high number of clusters (Var=95%).

From a qualitative viewpoint, an example of cluster extracted from the GTD (VAR=90%, MaxDensity-MIGRAL-CP, h=5) has the following centroid: [attack: 0.593; fire: 0.371; claim: 0.271; suspect: 0.1; report: 0.057; injur: 0.057; islam: 0.05] which allows us to identify a specific type of crime (terrorist attack). In Figure 4, it is possible to observe an example of a discovered criminal

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

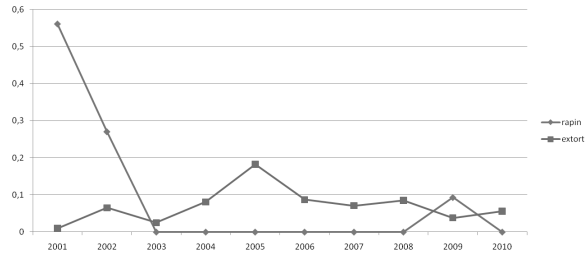


Fig. 4. A cluster evolution in the synthetic dataset. TF-IDF values are plotted.

evolution. Criminals belonging to a specific cluster in the first time window are tracked in the subsequent time windows. In the figure we plot the average weight of two relevant terms associated to the cluster centroid the criminals belong to. As we can see, while the weight of the term "rapin" decreases over time, the weight of the term "extort" increases and shows a peak in the period 2004-2006.

5 Conclusions

In this paper, we propose a framework which is able to incrementally extract knowledge from time-stamped news. The framework is four stepped: 1) it extracts features (terms) from news, 2) it represents subjects to which news are associated with, in terms of their semantic position, 3) it clusters together subjects and 4) it analyzes the clusters evolution. The framework has been evaluated in the context of risk identification and analysis in order to understand the evolution of criminal behaviors. Results on both real and synthetically generated datasets show that the algorithm for the identification of the criminal's semantic position provides the best results, independently of the considered configuration. For future works we intend to analytically identify the value of σ, with respect to h, such that the global optimum is guaranteed and to perform a more detailed qualitative evaluation on the evolutions discovered on real datasets. Finally, we intend to analyze how results may vary with different sizes of time windows.

References

1. Agarwal, N., Galan, M., Liu, H., Subramanya, S.: Wiscoll: Collective wisdom based blog clustering. Inf. Sci. **180** (January 2010) 39–61
2. Aggarwal, C.C.: On change diagnosis in evolving data streams. IEEE Trans. Knowl. Data Eng. **17**(5) (2005) 587–600
3. Allan, J., ed.: Topic Detection and Tracking: Event-based Information Organization. Kluwer International Series on Information Retrieval. Kluwer (2002)
4. Bansal, N., Chiang, F., Koudas, N., Tompa, F.W.: Seeking stable clusters in the blogosphere. In: VLDB, ACM (2007) 806–817
5. Brants, T., Chen, F., Farahat, A.: A system for new event detection. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. SIGIR '03, ACM (2003) 330–337

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

6. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '10, New York, NY, USA, ACM (2010) 333–342
7. Ceci, M., Appice, A., Malerba, D.: Time-slice density estimation for semantic-based tourist destination suggestion. In: ECAI. (2010)
8. Chau, M., Xu, J.J., Chen, H.: Extracting meaningful entities from police narrative reports. In: Proc. of the national conference on Digital government research. dg.o '02, Digital Government Society of North America (2002) 1–5
9. Chen, H., Chung, W., Xu, J., Wang, G., Qin, Y., Chau, M.: Crime data mining: A general framework and some examples. IEEE Computer **37** (2004) 50–56
10. Chung, S., Mcleod, D.: Dynamic Pattern Mining: An Incremental Data Clustering Approach. (2005) 85–112
11. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Mach. Learn. **42** (January 2001) 143–175
12. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of 2nd International Conference on Knowledge Discovery and. (1996) 226–231
13. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: NIPS. (2005)
14. Jolliffe, I.T.: Principal Component Analysis. Second edn. Springer (October 2002)
15. Jonas, J., Harper, J.: Effective Counterterrorism and the Limited Role of Predictive Data Mining. Policy Analysis **584** (December 2006)
16. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '02, New York, NY, USA, ACM (2002) 91–101
17. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: KDD '09, New York, NY, USA, ACM (2009) 497–506
18. Li, X., Yan, J., Fan, W., Liu, N., Yan, S., Chen, Z.: An online blog reading system by topic clustering and personalized ranking. ACM Trans. Internet Technol. **9** (July 2009) 9:1–9:26
19. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23) (2006) 8577–8582
20. Ozgul, F., Bondy, J., Aksoy, H.: Mining for offender group detection and story of a police operation. In: Proceedings of the sixth Australasian conference on Data mining and analytics - Volume 70. AusDM '07, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2007) 189–193
21. Schroeder, J., Xu, J., Chen, H., Chau, M.: Automated criminal link analysis based on domain knowledge: Research articles. J. Am. Soc. Inf. Sci. Technol. **58** (2007)
22. Seifert, J.W.: Data Mining and Homeland Security: An Overview. Bibliographisches Institut AG, Mannheim, W. Germany, Germany (2010)
23. Varlamis, I., Vassalos, V., Palaios, A.: Monitoring the evolution of interests in the blogosphere. In: Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on. (april 2008) 513 –518
24. Xu, J.J., Chen, H.: Fighting organized crime: Using shortest-path algorithms to identify associations in criminal networks. Decis. Support Syst. **38** (2004) 473–487
25. Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B., Liu, X.: Learning approaches for detecting and tracking news events. Intelligent Systems and their Applications, IEEE **14**(4) (1999) 32 –43
26. Zhong, S.: Efficient streaming text clustering. Neural Networks **18**(5-6) (2005)
27. Zhu, Y., Shasha, D.: Efficient elastic burst detection in data streams. In: ACM SIGKDD. KDD '03, New York, NY, USA, ACM (2003) 336–345

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Privacy-preserving Mining of Association Rules from Outsourced Transaction Databases*

Fosca Giannotti[1], Laks V.S. Lakshmanan[2], Anna Monreale[1,3], Dino Pedreschi[3], and Hui Wang[4]

[1] ISTI-CNR, Pisa, Italy, fosca.giannotti@isti.cnr.it
[2] University of British Columbia, Vancouver, Canada laks@cs.ubc.ca
[3] University of Pisa, Pisa, Italy, annam@di.unipi.it, pedre@di.unipi.it
[4] Stevens Institute of Technology, NJ, USA hwang@cs.stevens.edu

Abstract. Spurred by developments such as cloud computing, there has been considerable recent interest in the paradigm of data mining-as-service. A company (data owner) lacking in expertise or computational resources can outsource its mining needs to a third party service provider (server). However, both the items and the association rules of the outsourced database are considered private property of the corporation (data owner). To protect corporate privacy, the data owner transforms its data and ships it to the server, sends mining queries to the server, and recovers the true patterns from the extracted patterns received from the server. In this paper, we study the problem of outsourcing the association rule mining task within a corporate privacy-preserving framework. We propose a scheme for privacy preserving outsourced mining and show that the owner can recover the true patterns as well as their support by maintaining a compact synopsis.

1 Introduction

In recent years, there has been considerable interest in the data mining-as-service paradigm for enabling organizations with limited computational resources and/or data mining expertise to outsource their data mining needs to a third party service provider [2, 9, 6, 5, 13]. As an example, the operational transactional data from various outlets of Safeway, a grocery chain operating in the US and Canada, can be shipped to a third party which provides mining service for Safeway. The Safeway management need not employ an in-house team of data mining experts. Besides, they can cut down their local data management requirements because periodically data is shipped to the service provider who is in charge of maintaining it and conducting mining on it in response to requests from Safeway's business analysts. In this example, Safeway, the *client*, is a data *owner* and the service provider is referred to as the *server*. One of the main issues with this paradigm is that the server has access to valuable data of the owner and may learn sensitive information from it. E.g., by looking at the transactions, the server (or an intruder who gains access to the server) can learn which items are co-purchased, and in turn, the mined patterns. However, *both the transactions and the mined patterns*

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

are the property of Safeway and should remain safe from the server. This problem of protecting important private information of organizations/companies is referred to as "*corporate privacy*" [7]. Unlike *personal privacy*, which only considers the protection of the personal information recorded about individuals, corporate privacy requires that *both the individual items and the patterns of the collection of data items are regarded as corporate assets and thus must be protected.*

In this paper, we study the problem of outsourcing the association rule mining task within a corporate privacy-preserving framework (already presented in [8]). We propose

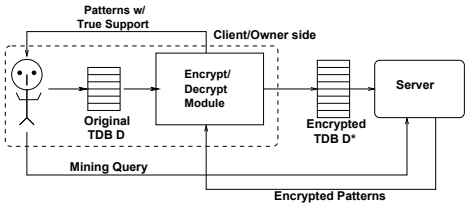


Fig. 1. Architecture of Mining-as-Service Paradigm.

an encryption scheme which enables privacy guarantees, and show some preliminary results obtained applying this model over large-scale, real-life transaction databases. The architecture behind our model is illustrated in Fig. 1. The client/owner encrypts its data using an encrypt/decrypt (ED) module, which can be treated as a "black box" from its perspective. This module is responsible for transforming the input data into an encrypted database. The server conducts data mining and sends the (encrypted) patterns to the owner. Our encryption scheme has the property that the returned supports are not true supports. The ED module recovers the true identity of the returned patterns as well their true supports.

2 Related Work

In this section we outline the work on privacy-preserving data publishing and mining.
Privacy-preserving data publishing (PPDP): The idea is that data is published with appropriate suppression, generalization, distortion, and/or decomposition such that individual privacy is not compromised and yet the published data is useful for mining [12, 10, 14].
Privacy-preserving data mining (PPDM): The main model here is that private data is collected from a number of sources by a collector for the purpose of consolidating the data and conducting mining. The collector is not trusted, so data is subjected to a random perturbation as it is collected. This body of work was pioneered by [2] and has been followed up by several papers since [11, 3].
Privacy-preserving pattern publishing (PPPP): The central question is how to publish results of mining such as frequent patterns without revealing any sensitive information about the underlying data [4], but the resulting patterns are disclosed.

A key distinction between our problem and the above mentioned PPDM problems is that, in our setting, not only the underlying data but also the mined results are not

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

intended for sharing and must remain private. The work that is most related to ours is [13]. Similar to our work, first, they utilize a one-to-n item mapping together with non-deterministic addition of cipher items to protect the identification of individual items. Second, they assume that the adversary may possess some prior knowledge of frequency of the itemsets, which can be used to decipher the encrypted items. While our attack model focuses on single items with the assumption that the attacker knows the exact frequency of every single item. The major issue left open by [13] is a formal protection result: their security analysis is entirely conducted empirically on various synthetic datasets.

3 Preliminaries: pattern mining

The reader is assumed to be familiar with the basics of association rule mining. Two major steps in mining association rules are: (i) finding frequent patterns and (ii) com-

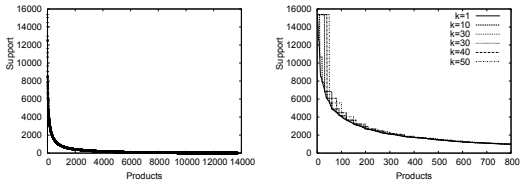


Fig. 2. Item Support Distribution: *CoopProd* (left); Encrypted *CoopProd* (right)

puting the association rules from them. Step (i) is the computationally dominant step. We briefly review frequent pattern mining below. Let $\mathscr{I} = i_1, ..., i_n$ be the set of items and $D = t_1, \ldots, t_m$ a transaction database (TDB) of transactions, each of which is a set of items. We denote the support of an itemset $S \subseteq \mathscr{I}$ as $supp_D(S)$ and the frequency by $freq_D(S)$. Recall, $freq_D(S) = supp_D(S)/|D|$. For each item i, $supp_D(i)$ and $freq_D(i)$ denote respectively the individual support and frequency of i. The whole function $supp_D(.)$, projected over items, is also called the *item support table* of D. It can be either represented in tabular form (see, e.g., Table 1 (a)), or plotted as a histogram, as in Fig. 2, where the item support distribution of the real-life Coop TDB is reported; both in support tables and in histograms items are listed in decreasing order of their support. The length of a transaction $t \in D$ is the number of items in t. We define the *size* of a TDB D as the sum of lengths of its transactions, i.e., $||D|| = \sum_{t \in D} |t|$. It is easy to see that $||D|| = \sum_{i \in \mathscr{I}} supp_D(i)$. This corresponds to the area under the support distribution graph (e.g., see Fig. 2). The frequent pattern mining problem [1] is: given a TDB D and a support threshold σ, find all patterns whose support in D is at least σ. We study a (corporate) privacy-preserving outsourcing framework for frequent pattern mining.

4 Privacy Model

We let D denote the original TDB that the owner has. To protect the identification of individual items, the owner applies an encryption function to D and transforms it to

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

D^*, the encrypted database. We refer to items in D as *plain items* and items in D^* as *cipher items*. The term item shall mean plain item by default. The notions of plain item sets, plain transactions, plain patterns, and their cipher counterparts are defined in the obvious way. We use $\mathscr{I}$ to denote the set of plain items and $\mathscr{E}$ to refer to the set of cipher items.

Adversary Knowledge. The server or an intruder (*attacker*) who gains access to the database may possess some background knowledge using which they can conduct attacks on the encrypted database D^* in order to make inferences.

We adopt a conservative model and assume that the attacker knows exactly the set of (plain) items $\mathscr{I}$ in the original transaction database D and their true supports in D, i.e., $supp_D(i), i \in \mathscr{I}$. The attacker may have access to similar data from a competing company, may read published reports, etc. Moreover we assume the attacker has access to the encrypted database D^*. Thus, he also knows the set of cipher items and their support in D^*, i.e., $supp_{D^*}(e)$, $e \in \mathscr{E}$.

In this paper we propose an encryption scheme based on: (i) replacing each plain item in D by a 1-1 substitution cipher (ii) adding fake transactions to the database. In particular, no new items are added. We assume the attacker knows this and thus he knows that $|\mathscr{E}| = |\mathscr{I}|$. We also assume the attacker knows the details of our encryption algorithm.

Attack Model. The data owner (i.e., the corporate) considers the true identity of: (1) every cipher item, (2) every cipher transaction, and (3) every cipher frequent pattern as the intellectual property which should be protected. If the cipher items are broken, i.e., their true identification is inferred by the attacker, then clearly cipher transactions and cipher patterns are broken, so they also must remain protected. The attack model is two-fold:

- *Item-based attack*: $\forall$ cipher item $e \in \mathscr{E}$, the attacker constructs a set of candidate plain items $Cand(e) \subset \mathscr{I}$. The probability that the cipher item e can be broken $prob(e) = 1/|Cand(e)|$.
- *Set-based attack*: Given a cipher itemset E, the attacker constructs a set of candidate plain itemsets $Cand(E)$, where $\forall X \in Cand(E)$, $X \subset \mathscr{I}$, and $|X| = |E|$. The probability that the cipher itemset E can be broken $prob(E) = 1/|Cand(E)|$.

We refer to $prob(e)$ and $prob(E)$ as *probabilities of crack*. From the point of view of the owner, minimizing the probabilities of crack is desirable. Intuitively, $Cand(e)$ and $Cand(E)$ should be as large as possible. Ideally, $Cand(e)$ should be the whole set of plaintext items. This can be achieved if we bring each cipher item to the same level of support, e.g., to the support of the most frequent item in D. Unfortunately, this option is impractical. In fact, we have a large increase in the size of D^* compared to D, i.e., a large size of the fake transactions. This in turn leads to a dramatic explosion of the frequent patterns, making pattern mining at the server side computationally prohibitive. This is the motivation for relaxing the equal-support constraint and introducing k-anonymity as a compromise.

Definition 1 (Item k-anonymity). *Let D be a transaction database and D^* its encrypted version. We say D^* satisfies the property of* item k-anonymity *provided for every*

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

cipher item $e \in \mathscr{E}$*, there are at least* $k-1$ *other distinct cipher items* $e_1, ..., e_{k-1} \in \mathscr{E}$ *such that* $supp_{D^*}(e) = supp_{D^*}(e_i)$*,* $1 \leq i \leq k-1$. □

Fig.2 (right) shows the effect of grouping together cipher items into groups of k items. For a given value of k, the support distribution resembles a descending staircase. With small k, the graph tends to the original support distribution in D; while as k increases, the graph gets closer to the horizontal line discussed above. As the size of D^* is the area below the graph, we can control the size of D^* by an appropriate choice of k.

To quantify the privacy guarantee of an encrypted database, we define the following notion:

Definition 2 (k–Privacy). *Given a database D and its encrypted version* D^**, we say* D^* *is* k-private *if: (1) for each cipher item* $e \in D^*$*,* $prob(e) \leq 1/k$*; and (2) for each cipher itemset E with support* $supp_{D^*}(E) > 0$*,* $prob(E) \leq 1/k$. □

This definition does not constrain the crack probability of cipher itemsets which have no support in D^*. Intuitively, such cipher itemsets are not interesting. This will be exploited in Sec. 5 in designing effective k-private encryption schemes. Formally, the problem we study is the following: Given a plain database D, construct a *k-private* cipher database D^* by using substitution ciphers and adding fake transactions such that from the set of frequent cipher patterns and their support in D^* sent to the owner by the server, the owner can reconstruct the true frequent patterns of D and their exact support. Additionally, we would like to minimize the space and time incurred by the owner in the process and the mining overhead incurred by the server.

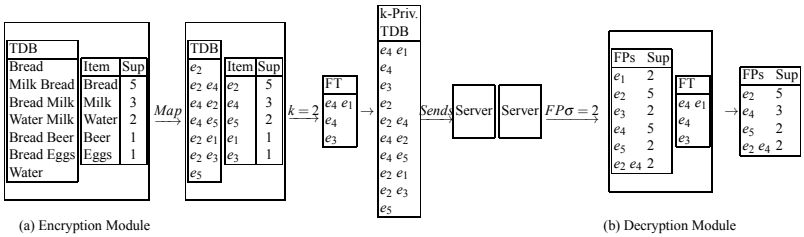


Fig. 3. E/D Module

5 Encryption/Decryption Scheme

In this section, we describe the ED module, originally presented in [8], responsible for the encryption of TDB and for the decryption of the cipher patterns coming from the server. The general idea of our Encryption/Decryption method is shown with an example in Figure 3.

5.1 Encryption

In this section, we introduce the encryption scheme, which transforms a TDB D into its encrypted version D^*. Our scheme is parametric w.r.t. $k > 0$ and consists of three main

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

steps: (1) using 1-1 substitution ciphers for each plain item; (2) using a specific item k-grouping method; (3) using a method for adding new fake transactions for achieving k-privacy. The encryption scheme is a countermeasure to the item-based and set-based attacks discussed in Sec. 4: since the attacker knows the exact support of each item, we create a k-private D^*, such that the cipher items cannot be broken based on their support.

k-Grouping Method. Given the items support table, several strategies can be adopted to cluster the items into groups of k. We assume the item support table is sorted in descending order of support and refer to cipher items in this order as e_1, e_2, etc. To obtain the formal protection that itemsets (or transactions) cannot be cracked with a probability higher than $\frac{1}{k}$, we need to use only grouping methods that yield groups of items that are *unsupported in D*. We call such grouping methods *robust*:

Definition 3. *Given a TDB D and a grouping G of the items occurring in D, G is called robust for D iff, for any group G_i of G, $supp_D(G_i) = 0$.* □

The above definition directly suggests a procedure for checking whether a given grouping G for a TDB D is robust: it is sufficient to check that the support in D of each group G_i in G is 0. If this is the case, the grouping can be safely used to obtain the maximum privacy protection guaranteed by our method. The following definition introduces our grouping method.

Definition 4. *Given the TDB D and its item support table in decreasing order of support, our grouping method:*

STEP1: groups together cipher items into groups of k adjacent items starting from the most frequent item e_1, obtaining the grouping $G = (G_1, ..., G_m)$.
STEP2: modifies the groups of G by repeating the following operations, until no group of items is supported in D: (1)Select the smallest $j \geq 1$ such that $supp_D(G_j) > 0$; (2) Find the most frequent item $i' \notin G_j$ such that, for the least frequent item i of G_j we have: $supp_D(G_j \setminus \{i\} \cup \{i'\}) = 0$; (3) Swap i with i' in the grouping. □

The output of grouping can be represented as the *noise table*. It extends the item support table with an extra column *Noise* indicating, for each cipher item e, the difference among the support of the most frequent cipher item in e's group and the support of e itself, as reported in the item support table. We denote the noise of a cipher item e as $N(e)$. The noise column indicates, for each cipher item e, the number of occurrences of e that are needed in D^* in order to bring e to the same support as the most frequent item of e's group. As such, the noise table represents the tool for generating the fake transactions to be added to D to obtain D^*. In particular, the total size of the needed fake transactions is exactly the summation of all the values in the Noise column of the noise table.

The noise table provides a compact *synopsis* (using $O(n)$ space, where n is the number of items) that can be stored by the ED module, to support both the creation of the fake transaction and the decryption step. For example, consider the example TDB in Figure 3, and its associated (cipher) item support table in Table 1 (a). For $k = 2$, the grouping method generates two groups: $\{e_2, e_5\}$ and $\{e_4, e_1, e_3\}$ (Table 1 (b)), that is

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

(a) IST

Item	Support
e_2	5
e_4	3
e_5	2
e_1	1
e_3	1

(b) Grouping

Item	Support
e_2	5
e_5	2
e_4	3
e_1	1
e_3	1

(c) Noise Table

Item	Support	Noise
e_2	5	0
e_5	2	3
e_4	3	0
e_1	1	2
e_3	1	2

(d) Hash Tables

	Table1
0	$\langle e_5, 1, 2\rangle$
1	$\langle e_3, 2, 0\rangle$

	Table2
0	$\langle e_1, 2, 0\rangle$

Table 1. Encryption with $k = 2$

robust: none of the two groups, considered as itemsets, is supported by any transaction in D.

Fake Transactions. Given a noise table specifying the noise $N(e)$ needed for each cipher item e, we generate the fake transactions as follows. First, we drop the rows with zero noise, corresponding to the most frequent items of each group or to other items with support equal to the maximum support of a group. Second, we sort the remaining rows in descending order of noise. Let $e'_1, \ldots, e'_m$ be the obtained ordering of (remaining) cipher items, with associated noise $N(e'_1), \ldots, N(e'_m)$. The following fake transactions are generated:

- $N(e'_1) - N(e'_2)$ instances of the transaction $\{e'_1\}$
- $N(e'_2) - N(e'_3)$ instances of the transaction $\{e'_1, e'_2\}$
- $\ldots$
- $N(e'_{m-1}) - N(e'_m)$ instances of the transaction $\{e'_1, \ldots, e'_{m-1}\}$
- $N(e'_m)$ instances of the transaction $\{e'_1, \ldots, e'_m\}$

Continuing the example, we consider cipher items of non-zero noise in Table 1 (c). The following two fake transactions are generated: 2 instances of the transaction $\{e_5, e_3, e_1\}$ and 1 instance of the transaction $\{e_5\}$. We observe that fake transactions introduced by this method may be longer than any transactions in the original TDB D. So, we consider shortening the lengths of the added fake transactions so that they are in line with the transaction lengths in D. In our running examples above, we obtain $\{e_5, e_3\}$, 2 of $\{e_1\}$ and 1 instance of $\{e_5\}$.

To implement the synopsis efficiently we use a hash table generated with a *minimal perfect hash function*. Minimal perfect hash functions are widely used for memory efficient storage and fast retrieval of items from static sets. In our scheme, the items of the noise table e_i with $N(e_i) > 0$ are the keys of the minimal perfect hash function. Given e_i, function h computes an integer in $[0 \ldots n-1]$, denoting the position of the hash table storing the triple of values $\langle e_i, times_i, occ_i\rangle$, where: $times_i$ represents the number of times the fake transaction $\{e_1, e_2, \ldots, e_i\}$ occurs in the set of fake transactions and occ_i is the number of times that e_i occurs altogether in the future fake transactions after the transaction $\{e_1, e_2, \ldots, e_i\}$.

Given a noise table with m items with non-null noise, our approach generates hash tables for the group of items. In general, the i-th entry of a hash table HT containing the item e_i has $times_i = N(e_i) - N(e_{i+1}), occ_i = \sum_{j=i+1}^{g} N(e_j)$, where g is the number of items in the current group. Notice that each hash table HT represents concisely the fake transactions involving all and only the items in a group of $g \leq l_{max}$ items. The hash tables for the items of non-zero noise in Table 1 (c) are shown in Table 1(d). Given that in our example, $l_{max} = 2$, we need to split the 3 items of non-zero noise in Table 1 into two sets, each with associated fake transactions, coded by the two hash tables.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Notice that any pattern consisting of items from different hash tables is *not* supported by any fake transactions. Finally, we use a (second-level) ordinary hash function H to map each item e to the hash table HT containing e.

The constructed fake transactions are added to D (once items are replaced by cipher items) to form D^*, and transmitted to the server. A record of the fake transactions, i.e., $DF = D^* \setminus D$, is stored by the ED module, by the compact synopsis described above.

5.2 Decryption

When the client requests the execution of a pattern mining query to the server, specifying a minimum support threshold σ, the server returns the computed frequent patterns from D^*. Clearly, for every itemset S and its corresponding cipher itemset E, we have $supp_D(S) \leq supp_{D^*}(E)$. Therefore, our encryption scheme guarantees that all itemsets frequent in D will be returned, in cipher version, by the server. But additional patterns frequent in D^*, but not in D, are returned as well. For each cipher pattern E returned by the server together with $supp_{D^*}(E)$, the ED module trivially recovers the corresponding plain pattern S as follows: $supp_D(S) = supp_{D^*}(E) - supp_{D^* \setminus D}(E)$.

This calculation is efficiently performed by the ED module using the synopsis of the fake transactions in $D^* \setminus D$ described above.

6 Preliminary Experimental Results

Data Set: We empirically assess our encryption method with respect to a real-life transaction database donated by Coop, a cooperative of consumers that is today the largest supermarket chain in Italy[5]. We selected 300,000 transactions involving 13,730 different *products*.

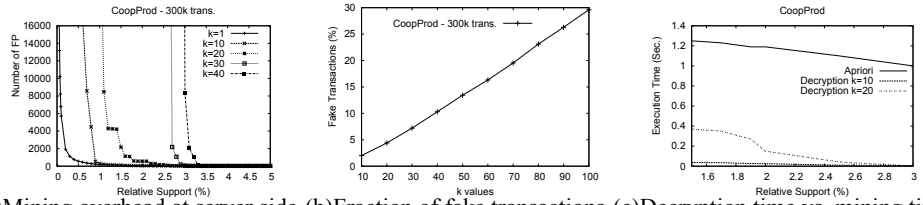
Encryption overhead: we assess the size of fake transactions added to *CoopProd** after encryption; Fig. 4 (b) reports the sizes of fake transactions for different k values. We observe that the size of fake transactions increases linearly with k.

Mining overhead: We study the overhead at server side in the pattern mining task over *CoopProd** w.r.t. *CoopProd*. We adopted the *Apriori* implementation by Christian Borgelt[6]. Instead of measuring performance in run time, we measure the increase in the number of frequent patterns obtained from mining the encrypted TDB, considering different support thresholds. Results are plotted in Fig. 4(a), for different values of k; notice that $k = 1$ means that the original and encrypted TDB are the same. We observe that the number of frequent patterns, at a given support threshold, increases with k, as expected. However, mining over *CoopProd** exhibits a small overhead even for very small support thresholds, e.g., a support threshold of about 1% for $k = 10$ and 1.5% for $k = 20$. We found that, for reasonably small values of the support threshold, the incurred overhead at server side is kept under control; clearly, a trade-off exists between the level of privacy, which increases with k, and the minimum affordable support threshold for mining, which also increases with k.

[5] http://www.e-coop.it/, in Italian

[6] http://www.borgelt.net

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,



(a)Mining overhead at server side (b)Fraction of fake transactions (c)Decryption time vs. mining time

Fig. 4. Overhead at server side and Decryption overhead

Decryption overhead by the ED module: We now consider the feasibility of the proposed outsourcing model. The ED module encrypts the TDB once which is sent to the server. Mining is conducted repeatedly at the server side and decrypted every time by the ED module. Thus, we need to compare the decryption time with the time of directly executing apriori over the original database. As shown in Fig. 4(c), the decryption time is about one order of magnitude smaller than the mining time; for higher support threshold, the gap increases to about two orders of magnitude.

7 Conclusion and Future Work

We studied the problem of (corporate) privacy-preserving outsourcing of association rule mining. Our encryption scheme is based on 1-1 substitutions and fake transactions such that the transformed database satisfies k-anonymity w.r.t. items and itemsets. Moreover we showed some preliminary empirical results that are encouraging; naturally, there are many interesting open issues to be investigated. The next steps include: (1) The study of a formal analysis based on our attack model and the proof that the probability that an itemset can be broken by the server can always be controlled to be below a threshold chosen by the owner, by setting the anonymity threshold k; (2) The complexity analysis of our encryption/decryption scheme; (3) The definition of an strategy for incrementally maintaining the synopsis at the client side against updates in the form of appends. (4) scheme using large real data set with different sparsity/density properties to understand how it works in different settings; (5) The analysis of the scalability of the proposed approach and comparison of the execution time of the mining step with that of the decryption step by using a mining algorithm like FP-growth, that could be more efficient than an apriori-based algorithm.

References

1. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. VLDB 1994.
2. R. Agrawal and R. Srikant. Privacy-preserving data mining. SIGMOD 2000.
3. S. Agrawal, J. R. Haritsa: A Framework for High-Accuracy Privacy-Preserving Mining. ICDE 2005.
4. M. Atzori, F. Bonchi, F. Giannotti, D. Pedreschi: Anonymity preserving pattern discovery. VLDB J. 17(4): 703-727 (2008).

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

5. S. Bu et al. Preservation of patterns and input-output privacy. ICDE 2007
6. K. Chen et al. Toward attack-resilient geometric data perturbbation. SDM, 2007.
7. C. Clifton, M. Kantarcioglu, J. Vaidya. Defining Privacy for Data Mining. NSF Workshop on Next Generation Data Mining, 2002.
8. F. Giannotti, L.V.S. Lakshmanan, A. Monreale, D. Pedreschi and H. Wang. Privacy-Preserving Data Mining from Outsourced Databases. CPDP 2011.
9. L.V.S. Lakshmanan, R.T. Ng, G. Ramesh. To Do or Not To Do: The Dilemma of Disclosing Anonymized Data. SIGMOD 2005.
10. A. Machanavajjhala, J. Gehrke, and D. Kifer. l-diversity: Privacy beyond k-anonymity. In ICDE, 2006.
11. S. Rizvi, J. R. Haritsa: Maintaining Data Privacy in Association Rule Mining. VLDB 2002.
12. P. Samarati. Protecting respondents' identities in microdata release. TKDE, 13(6):1010-1027, 2001.
13. W. K. Wong, D. W. Cheung, E. Hung, B. Kao, N. Mamoulis. Security in Outsourcing of Association Rule Mining. VLDB 2007.
14. X. Xiao and Y. Tao. Anatomy: Simple and Effective Privacy Preservation. VLDB 2006.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Frequent Itemset Mining of Distributed Uncertain Data under User-Defined Constraints*

Alfredo Cuzzocrea[1] and Carson K. Leung[2]

[1] ICAR-CNR and University of Calabria, Italy
cuzzocrea@si.deis.unical.it
[2] University of Manitoba, Canada
kleung@cs.umanitoba.ca

Abstract. Many existing *distributed data mining* algorithms do not allow users to express the patterns to be mined according to their intention via the use of constraints. Consequently, these unconstrained mining algorithms can yield numerous patterns that are not interesting to users. Moreover, due to inherited measurement inaccuracies and/or network latencies, data are often riddled with uncertainty. These call for *constrained mining* and *uncertain data mining*. In this paper, we propose a tree-based system for mining frequent itemsets that satisfy user-defined constraints from a distributed environment such as a wireless sensor network of uncertain data.

Keywords: Data mining, knowledge discovery, constraints, distributed data, frequent itemsets, uncertain data.

1 Introduction and Related Work

Many frequent itemset mining algorithms in the early days were Apriori-based [1, 19], which depends on a generate-and-test paradigm to find all frequent itemsets by first generating candidates and then checking their support (i.e., their occurrences) against traditional databases (DBs) containing precise data. To avoid the generate-and-test paradigm, the FP-growth algorithm [7] was proposed. Such a tree-based algorithm constructs a Frequent Pattern tree (FP-tree) to capture the contents of the DBs; it focuses on frequent pattern growth which is a restricted test-only approach.

In many real-life applications, data are riddled with uncertainty. It is partially due to inherent measurement inaccuracies, sampling and duration errors, network latencies, and intentional blurring of data to preserve anonymity. As such, the presence or absence of items in a DB is uncertain. Hence, mining uncertain data [2, 11, 12, 15, 21] is in demand. For example, a physician may highly suspect (but not guarantee) that a patient suffers from asthma. The uncertainty of such suspicion can be expressed in terms of *existential probability* $P(x, t_i)$ of an item x in a transaction t_i in a probabilistic DB. To mine frequent itemsets from these uncertain data, the UF-growth [13] algorithm was proposed.

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Many frequent itemset mining algorithms, regardless whether Apriori-based or tree-based, provide little or no support for user focus when mining precise or uncertain data. However, in many real-life applications, the user may have some particular phenomena in mind on which to focus the mining (e.g., medical analysts may want to find only those lab test records belonging to patients suspected to suffer from asthma instead of all the patients). Without user focus, the user often needs to wait for a long period of time for numerous frequent itemsets, out of which only a tiny fraction may be interesting to the user. This calls for *constrained frequent itemset mining* [9, 14], which finds frequent itemsets that satisfy user-defined constraints. For example, DCF [8] mines constrained frequent itemsets from traditional precise data.

With advances in technology, one can easily collect large amounts of data from not only a single source but multiple sources. For example, in recent years, sensor networks have been widely used in many application areas such as agricultural, architectural, environmental, and structural surveillance. Sensors distributed in these networks serve as good sources of data. However, sensors usually have limited communication bandwidth, transmission energy, and computational power. Thus, data are not usually transmitted to a single distant centralized processor to perform the data mining task. Instead, data are transmitted to their local (e.g., closest) processors within a distributed environment. This calls for *distributed mining* [4, 16–18, 20]—which discovers implicit, previously unknown, and potentially useful knowledge that might be embedded in distributed data.

Existing distributed mining algorithms—such as FDM [3] and Parallel-HFP-Leap [6]—find frequent itemsets in a distributed environment, but they all do not handle constraints nor do they mine uncertain data. In contrast, DCF finds *constrained* frequent itemsets, but they mine a centralized DB of precise data. UF-growth mines a centralized DB of *uncertain* data for all (unconstrained) frequent itemsets instead of only those constrained ones. In other words, these existing frequent itemset mining algorithms fall short in different aspects. Hence, a natural question to ask is: Is it possible to mine *uncertain* data for only those frequent itemsets that satisfy user *constraints* in a *distributed* environment? In response to this question, we propose in this paper a tree-based system for mining uncertain data in a distributed environment for frequent itemsets that satisfy user-defined constraints. Here, our *key contribution* is the non-trivial integration of (i) constrained mining, (ii) distributed mining, (iii) uncertain data mining, with (iv) tree-based frequent itemset mining. The resulting tree-based system efficiently mines from distributed uncertain data for only those constrained frequent itemsets.

This paper is organized as follows. In the next section, we propose our non-trivial integration of tree-based frequent set mining, constrained mining, distributed mining, and uncertain mining. Experimental results are shown in Section 3. Finally, Section 4 presents the conclusions.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2 Our Proposed Distributed Mining System

Without loss of generality, we assume to have p sites/processors and $m = m_1 + m_2 + ... + m_p$ sensors in a distributed network such that m_1 wireless sensors transmit data to their closest or designated site/processor P_1, m_2 sensors transmit data to the site/processor P_2, and so on. With this setting, our distributed mining system finds (i) constrained itemsets that are locally frequent w.r.t. site/processor P_i and (ii) those that are globally frequent w.r.t. all sites/processors in the entire wireless sensor network.

2.1 Finding Constrained Locally Frequent Itemsets

Given m_i sensors transmitting data to the processor P_i, a local database TDB_i of uncertain data can be created for P_i. Here, we use the "possible world" interpretation of uncertain data. We aim to find itemsets that are both (i) locally frequent to P_i and (ii) satisfying a *succinct* constraint, to which a majority of user-defined constraints belong. A constraint C is ***succinct*** [9] if one can directly generate precisely all and only those itemsets satisfying C without generating and excluding itemsets not satisfying C. Examples of succinct constraints include $C_1 \equiv max(X.Price) \leq \$25$ (which expresses the user interest in finding every itemset X such that the maximum price of all market basket items in X is at most \$25) and $C_2 \equiv min(X.Price) \leq \$30$ (which says that the minimum price of all items in X is at most \$30). Note that many non-succinct constraints (e.g., $C_3 \equiv avg(X.Price) \leq \$30$) can be induced into weaker but succinct constraints (e.g., C_2).

Step 1: Identification of items satisfying the constraint. A succinct constraint C is also *anti-monotone* [9] if all subsets of an itemset X satisfy C whenever X satisfies C. Hence, succinct constraints can be further into (i) ***succinct anti-monotone* (*SAM*)** and (ii) ***succinct non-anti-monotone* (*SUC*)** constraints. Then, let $\texttt{Item}^{\texttt{M}}$ be the collection of *mandatory items*—i.e., the collection of domain items that individually satisfy C (e.g., SAM or SUC constraint); let $\texttt{Item}^{\texttt{O}}$ be the collection of *optional items*—i.e., the collection of domain items that individually violate C.

For any C_{SAM}, an itemset X satisfying C_{SAM} cannot contain any item from $\texttt{Item}^{\texttt{O}}$ (e.g., if an itemset X containing an item having price $>$ \$25, then X violates C_{SAM} C_1 and so does every superset of X). So, any X satisfying C_{SAM} must be generated by combining items from $\texttt{Item}^{\texttt{M}}$ (i.e., $X \subseteq \texttt{Item}^{\texttt{M}}$). Items in $\texttt{Item}^{\texttt{M}}$ can be efficiently enumerated (from the list of domain items) by selecting only those items that individually satisfy C_{SAM}. Due to page limitation, please see Ref. [5] for an illustrative example.

For any C_{SUC}, any itemset X satisfying C_{SUC} is composed of mandatory items (i.e., items that individually satisfy C_{SUC}) and possibly some optional items (regardless whether or not they satisfy C_{SUC}). Note that, if X violates C_{SUC}, there is no guarantee that all or any of its supersets would violate C_{SUC}.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Hence, any itemset X satisfying C_{SUC} must be generated by combining at least one $\texttt{Item}^{\texttt{M}}$ item and possibly some $\texttt{Item}^{\texttt{O}}$ items. Due to succinctness, items in $\texttt{Item}^{\texttt{M}}$ and in $\texttt{Item}^{\texttt{O}}$ can be efficiently enumerated. See Ref. [5] for an illustrative example.

Step 2: Construction of an UF-Tree. Once the domain items are classified into $\texttt{Item}^{\texttt{M}}$ and $\texttt{Item}^{\texttt{O}}$ items (no $\texttt{Item}^{\texttt{O}}$ items for C_{SAM}), our system then constructs an UF-tree, which is built in preparation for mining constrained frequent itemsets from uncertain data. It does so by first scanning the DB of uncertain data once. It accumulates the *expected support* of each of the items in order to find all *frequent* domain items. The ***expected support*** [10] of an itemset X consisting of k independent items over n transaction in the DB can be computed by $expSup(X) = \sum_{i=1}^{n} \left(\prod_{x \in X} P(x, t_i) \right)$. The system discards infrequent items and only captures frequent items in the UF-tree. Note that any infrequent $\texttt{Item}^{\texttt{M}}$ or $\texttt{Item}^{\texttt{O}}$ items can be safely discarded because any itemset containing an infrequent item is also infrequent.

Once the frequent $\texttt{Item}^{\texttt{M}}$ and $\texttt{Item}^{\texttt{O}}$ items are found, our system arranges these two kinds of items in such a way that $\texttt{Item}^{\texttt{M}}$ items appear *below* $\texttt{Item}^{\texttt{O}}$ items (i.e., $\texttt{Item}^{\texttt{M}}$ items are closer to the leaves, and $\texttt{Item}^{\texttt{O}}$ items are closer to the root). Among all the items in $\texttt{Item}^{\texttt{M}}$, they are sorted in non-ascending order of accumulated expected support. Similarly, among all the items in $\texttt{Item}^{\texttt{O}}$, they are also sorted in non-ascending order of accumulated expected support. The system then scans the DB the second time and inserts each transaction of the DB into the UF-tree. Here, the new transaction is merged with a child (or descendant) node of the root of the UF-tree (at the highest support level) only if the same item *and the same expected support* exist in both the transaction and the child (or descendant) nodes.

For C_{SAM}, the corresponding UF-tree captures only those frequent $\texttt{Item}^{\texttt{M}}$ items; for C_{SUC}, the corresponding UF-tree captures both the frequent $\texttt{Item}^{\texttt{M}}$ items and the frequent $\texttt{Item}^{\texttt{O}}$ items. With such a tree construction process, the UF-tree possesses the property that *the occurrence count of a node is at least the sum of occurrence counts of all its child nodes.* For an illustrative example, see Ref. [5].

Step 3: Mining of Constrained Frequent Itemsets from the UF-Tree. Once the UF-tree is constructed with the item-ordering scheme where $\texttt{Item}^{\texttt{O}}$ items are above $\texttt{Item}^{\texttt{M}}$ items, our proposed system extracts appropriate paths to form a projected DB for each $x \in \texttt{Item}^{\texttt{M}}$. The system does not need to form projected DBs for any $y \in \texttt{Item}^{\texttt{O}}$ because all itemsets satisfying C_{SUC} must be "extensions" of an item from $\texttt{Item}^{\texttt{M}}$ (i.e., all valid itemsets must be grown from $\texttt{Item}^{\texttt{M}}$ items) and no $\texttt{Item}^{\texttt{O}}$ items are kept in the UF-tree for C_{SAM}.

When forming each $\{x\}$-projected DB and constructing its UF-tree, our system does not need to distinguish those $\texttt{Item}^{\texttt{M}}$ items from $\texttt{Item}^{\texttt{O}}$ items in the UF-tree for $\{x\}$-projected DB. Such a distinction between two kinds of items is only needed for the UF-tree for the DB (for SUC constraints only) but not

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

projected UF-trees once we found at least one valid item $x \in \texttt{Item}^{\texttt{M}}$ because, for any v satisfying C_{SUC}, $v = \{x\} \cup \texttt{others}$, where (i) $x \in \texttt{Item}^{\texttt{M}}$, (ii) $\texttt{others} \subseteq (\texttt{Item}^{\texttt{M}} \cup \texttt{Item}^{\texttt{O}} - \{x\})$. After constructing these projected UF-tree for each $x \in \texttt{Item}^{\texttt{M}}$, our proposed system mines all frequent itemsets that satisfy C_{SUC} in the same manner as it mines those satisfying C_{SAM}. Again, see Ref. [5] for an illustrative example.

2.2 Finding Constrained Globally Frequent Itemsets

Once the constrained locally frequent itemsets are found from distributed uncertain data, the next step is to find the constrained globally frequent itemsets among those constrained locally frequent itemsets. Note that it is not a good idea to transmit all data TDB_i from each site/processor P_i to a centralized site/processor Q, where all data are merged to form a global database $TDB = \bigcup_i TDB_i$ from which constrained globally frequent itemsets are found. The problem with such an approach is that it requires lots of communication for transmitting data from each site. This problem is worsen when TDB_i's are huge; wireless sensors can generate huge amount of data. Moreover, such an approach does not make use of constrained locally frequent itemsets in finding constrained globally frequent itemsets.

Similarly, it is also not a good idea to ask each site to transmit all its constrained locally frequent itemsets to a centralized site, where the itemsets are merged. The merge result is a collection of global candidate itemsets. The problem is that if a constrained itemset X is locally frequent at a site P_1 but not at another site P_2, then we do not have the frequency of X at P_2. Lacking this frequency information, one may not be able to determine whether X is globally frequent or not.

Instead, our proposed system does the following. Each site/processor P_i (for $1 \leq i \leq p$) applies constraint checking and frequency checking to find locally frequent $\texttt{Item}_i^{\texttt{M}}$ items (and $\texttt{Item}_i^{\texttt{O}}$ items for C_{SUC}), which are then transmitted to a centralized site/processor Q. It takes the union of these items, and broadcasts the union to all P_i's. Each P_i then extracts these items (potentially globally frequent items) from transactions in TDB_i and puts into an UF-tree. Note that all globally frequent itemsets must be composed of only the items from this union because: (i) if an item A is globally frequent, A must be locally frequent in at least one of P_i's; (ii) if an item B is locally infrequent in *all* the P_i's, B is guaranteed to be globally infrequent.) At each site P_i, the UF-tree contains (i) items that are locally frequent w.r.t. P_i and (ii) items that are potentially globally frequent but locally infrequent items w.r.t P_i. Then, our system recursively applies the usual tree-based mining process (e.g., UF-growth) to each α-projected DB (where locally frequent $\alpha \subseteq \texttt{Item}_i^{\texttt{M}}$) of the UF-tree at P_i to find *constrained locally frequent itemsets* (with local frequency information). These itemsets are then sent to Q, where the local frequencies are summed. As a result, *constrained globally frequent itemsets* can be found. If the sum of available local frequencies of a constrained itemset X meets the minimum support threshold, then X is globally frequent. For the case where a constrained itemset is locally frequent at

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

a site P_1 but not at another site P_2, then Q sends a request to P_2 for finding its local frequency. It is guaranteed that such frequency information can be found by traversing appropriate paths in the UF-tree at P_2 (because the UP-tree keeps all potential globally frequent items).

To summarize, given p sites/processors in a distributed environment (e.g., a wireless sensor network), our system makes use of (i) the constrained locally frequent itemsets and (ii) the UF-trees that keep all potentially global frequent items to efficiently find constrained globally frequent itemsets (w.r.t. the entire distributed environment). Again, constraints are pushed inside the mining process; the computation is proportional to the selectivity of constraints. Moreover, our proposed system does not require lots of communication among processors (e.g., it does not need to transmit TDB_i).

3 Experimental Results

For experimental evaluation, we used different datasets including IBM synthetic data, real-life DBs from the UC Irvine Machine Learning Depository as well as those from the Frequent Itemset Mining Implementation (FIMI) Dataset Repository. Due to page limitation, we cite below those experimental results based on a dataset generated by the program developed at IBM Almaden Research Center [1]. The dataset contains 10M records with an average transaction length of 10 items, and a domain of 1,000 items. Unless otherwise specified, we used *min-sup* = 0.01%. We randomly assigned to each item an existential probability in the range of (0,1]. All experiments were run in a time-sharing environment in a 2.4 GHz machine. The reported figures are based on the average of multiple runs. Runtime includes CPU and I/Os for constraint checking, UF-tree construction, and frequent itemset mining steps.

In the first experiment, we evaluated the accuracy and flexibility of our proposed system, which was implemented in C. For instance, we used (i) a dataset of uncertain data and (ii) a *constraint with 100% selectivity* (so that every item is selected). With this setting, we compared our system (which mines *constrained* frequent itemsets from uncertain data) with UF-growth [13] (which mines *unconstrained* itemsets). Experimental results showed that (i) our system is as accurate as UF-growth (and they both returned the *same* collection of frequent itemsets), and (ii) our system is more flexible than UF-growth (because the former is capable of finding frequent itemsets from *distributed* uncertain data with constraints of *any selectivity* whereas the latter is confined to those of 100% selectivity.

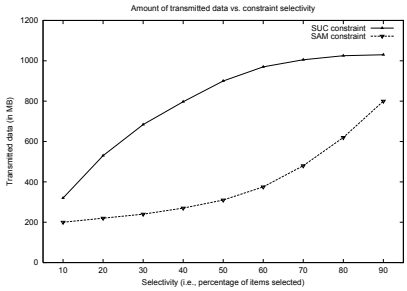
Similar observations were made when we compared our system (which mines *uncertain* data) with DCF [8] (which mines *precise* data) by using (i) a constraint and (ii) a dataset of uncertain data consisting of items *all with existential probability of 1* (indicating that all items are definitely present in the DB). We observed that (i) our system is as accurate as DCF, and (ii) our proposed system is more flexible than DCF (because the former is capable of finding frequent itemsets containing items with *various existential probability values* ranging from 0 to 1 whereas the latter is confined to those of existential probability of 1.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

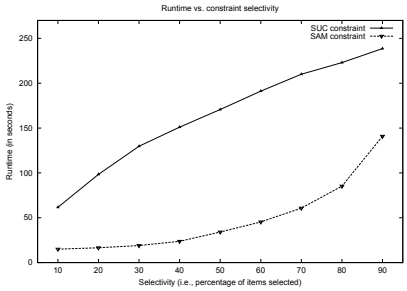
In the second experiment, we evaluated the *effectiveness of constrained mining in a distributed environment* by measuring the amount of communication/ data transmitted between the distributed sites P_i's and their centralized site Q. Fig. 1(a) shows that the amount of transmitted data decreased when the selectivity of constraints decreased. Fig. 1(b) shows the corresponding runtimes. Specifically, runtimes decreased when the selectivity of constraints decreased.

In the third experiment, we evaluated the effects of varying the number of distributed sites. When more sites were in the distributed network, our system transmitted more data because an addition of a site implies transmission of an additional set of locally frequent items and locally frequent itemsets. In terms of runtime, when more sites were in the network, the runtime of our system increased slightly. This is because the extra communication time (due to extra sites) was offset by the savings in building and mining from a smaller UF-tree at each site. For example, when we doubled the number of sites (from 4 to 8 sites), the amount of communication/data transmitted was almost double (because each site produced a similar set of locally frequent itemsets—especially when TDB_i's were similar); but, the runtime just increased slightly (1.07 times; i.e., not doubled) because we built and mined from smaller FP-trees at 8 sites (rather than from bigger trees at 4 sites).

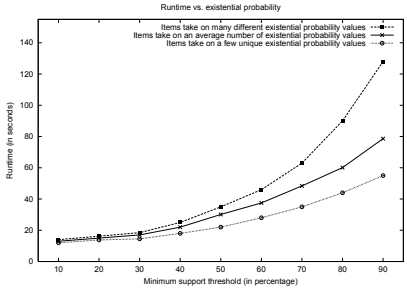
In addition, we conducted a few more experiments. For example, we tested the effect of distribution of existential probabilities of items. When items took



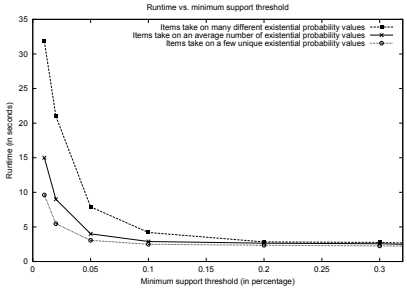
(a) Amt of transmitted data vs. selectivity



(b) Runtime vs. selectivity



(c) Runtime vs. existential probability



(d) Runtime vs. minsup

Fig. 1. Experimental results of our proposed system

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

on a few unique existential probability values, UF-trees became smaller and thus took shorter runtimes. See Fig. 1(c). We also tested the effect of *minsup*. When *minsup* increased, fewer itemsets had expected support $\geq$ *minsup*, and thus shorter runtimes were required for the experiments. See Fig. 1(d).

All these experimental results showed the importance and the benefits of using our proposed system in mining probabilistic datasets of uncertain data for frequent itemsets.

4 Conclusions

In this paper, we proposed a tree-based system for mining frequent itemsets that satisfy user-defined constraints from a distributed environment such as a wireless sensor network of uncertain data. Experimental results show effectiveness of our proposed system. As future work, we plan to extend our experiments to additional datasets.

Acknowledgement. This project is partially supported by NSERC (Canada) and University of Manitoba.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB 1994, pp. 487–499.
2. Bernecker, T., Kriegel, H.-P., Renz, M., Verhein, F., Zuefle, A.: Probabilistic frequent itemset mining in uncertain databases. In: ACM KDD 2009, pp. 119–127.
3. Cheung, D.W., Han, J., Ng, V.T., Fu, A.W., Fu, Y.: A fast distributed algorithm for mining association rules. In: PDIS 1996, pp. 31–42.
4. Coenen, F., Leng, P., Ahmed, S.: T-trees, vertical partitioning and distributed association rule mining. In: IEEE ICDM 2003, pp. 513–516.
5. Cuzzocrea, A., Leung, C.K.-S.: Distributed mining of constrained frequent sets from uncertain data. In: ICA3PP 2011, Part 1. LNCS 7016, pp. 40–53.
6. El-Hajj, M., Zaïane, O.R.: Parallel leap: large-scale maximal pattern mining in a distributed environment. In: ICPADS 2006, pp. 135–142.
7. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000, pp. 1–12.
8. Lakshmanan, L.V.S., Leung, C.K.-S., Ng, R.T.: Efficient dynamic mining of constrained frequent sets. ACM TODS 28(4), 337–389 (2003)
9. Leung, C.K.-S.: Frequent itemset mining with constraints. In: Encyclopedia of Database Systems, pp. 1179–1183 (2009)
10. Leung, C.K.-S.: Mining uncertain data. Wiley WIDM 1(4), pp. 316–329 (2011)
11. Leung, C.K.-S., Hao, B.: Mining of frequent itemsets from streams of uncertain data. In: IEEE ICDE 2009, pp. 1663–1670.
12. Leung, C.K.-S., Jiang, F., Hayduk, Y.: A landmark-model based system for mining frequent patterns from uncertain data streams. In: IDEAS 2011, pp. 249–250.
13. Leung, C.K.-S., Mateo, M.A.F., Brajczuk, D.A.: A tree-based approach for frequent pattern mining from uncertain data. In: PAKDD 2008. LNAI 5012, pp. 653–661.
14. Leung, C.K.-S., Sun, L.: A new class of constraints for constrained frequent pattern mining. In: ACM SAC 2012, pp. 518–523.
15. Leung, C.K.-S., Tanbeer, S.K.: Fast tree-based mining of frequent itemsets from uncertain data. In: DASFAA 2012, Part 1. LNCS 7238, pp. 272–287.
16. Park, J.S., Chen, M.-S., Yu, P.S.: Efficient parallel data mining for association rules. In: CIKM 1995, pp. 31–36.
17. Provost, F.: Distributed data mining: scaling up and beyond. In: Advances in Distributed and Parallel Knowledge Discovery, pp. 3–28 (2000)
18. Schuster, A., Wolff, R., Trock, D.: A high-performance distributed algorithm for mining association rules. Springer KAIS 7(4), pp. 458–475 (2005)
19. Toivonen, H.: Sampling large databases for association rules. In: VLDB 1996, pp. 134–145.
20. Zaki, M.J.: Parallel and distributed association mining: a survey. IEEE Concurrency 7(4), pp. 14–25 (1999)
21. Zhang, Q., Li, F., Yi, K.: Finding frequent items in probabilistic data. In: ACM SIGMOD 2008, pp. 819–832.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Hierarchical Latent Factors for Preference Data*

Nicola Barbieri[1,2], Giuseppe Manco[2], and Ettore Ritacco[2]

[1] Department of Electronics, Informatics and Systems - University of Calabria,
via Bucci 41c, 87036 Rende (CS) - Italy
nbarbieri@deis.unical.it,
[2] Institute for High Performance Computing and Networks (ICAR)
Italian National Research Council
via Bucci 41c, 87036 Rende (CS) - Italy
{barbieri,manco,ritacco}@icar.cnr.it

Abstract. In this work we propose a probabilistic hierarchical generative approach for users' preference data, which is designed to overcome the limitation of current methodologies in Recommender Systems and thus to meet both prediction and recommendation accuracy. The *Bayesian Hierarchical User Community Model (BH-UCM)* focuses both on modeling the popularity of items and the distribution over item ratings. An extensive evaluation over two popular benchmark datasets shows that the combined modeling of item popularity and rating provides a powerful framework both for rating prediction and for the generation of accurate recommendation lists.

1 Introduction

Recommender systems (RS) play an important role in several domains as they provide users with potentially interesting recommendations within catalogs of available information/products/services [11]. Recent studies [5, 6, 9] have shown that the focus on the prediction accuracy (e.g., in terms of root mean square error, RMSE) does not necessarily help in devising high quality recommendations. It has been shown [1] that probabilistic approaches based on latent-factor models allow the most adequate degree of flexibility, as they: *(i)* allow the specification of complex yet easy to interpret latent structures; *(ii)* achieve the highest recommendation accuracy.

In this paper we propose a new Bayesian Hierarchical latent factor model (*Bayesian Hierarchical User Community Model*, *BH-UCM* in the following) which combines the advantages of both hierarchical modeling and item selection, and comparatively investigate both its recommendation accuracy and prediction error. The underlying generative process takes into account both item selection and rating emission, so that those users who experience the same items and tend to adopt the same rating pattern are gathered into communities. Individual users are modeled as a random mixture of communities, where the individual community is characterized again by a mixture of topics modeling both the popularity of items and the distribution over item ratings.

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2 Preliminaries and Context

We introduce in this section the notation used throughout the paper along with some preliminary concepts. Let $\mathcal{U} = \{u_1, \ldots, u_M\}$ be a set of M users and $\mathcal{I} = \{i_1, \ldots, i_N\}$ a set of N items. Users' preferences can be represented as a $M \times N$ matrix $\mathbf{R}$, whose generic entry r_i^u denotes the rating value (i.e., the degree of preference) assigned by user u to item i. For each pair $\langle u, i \rangle$, rating value r_i^u falls within a limited integer range $\mathcal{V} = \{0, \ldots, V\}$, where 0 represents an unknown rating and V is the maximum degree of preference. The number of users M as well as the number of items N are very large and, in practical applications, the rating matrix $\mathbf{R}$ is characterized by an exceptional sparseness (e.g., more than 95%), since the individual users tend to rate a limited number of items.

Given an active user u, the goal of a RS is to provide u with a recommendation list of unexperienced items that are expected to be of interest to u. This clearly involves predicting the interest of u into unrated items.

Based on the underlying mathematical model, probabilistic approaches allow the prediction of the expected interest of a user u into an item i in two different ways [8]:

- *Forced prediction*: the probabilistic model provides an estimate of $P(r|u, i)$;
- *Free prediction*: the item selection process is included in the probabilistic model, which is typically based on the estimate of $P(r, i|u)$.

In general, a recommendation list can be generated by selecting a set of candidate items, sorting them according a ranking function which is provided by the RS and selecting the top k items.

3 Bayesian Hierarchical Model for Preference Data

The key idea of the proposed technique is that there exists a set of user communities, each one describing different tastes of users and their corresponding rating patterns. Each user community is then modeled as a random mixture over latent topics, which can be interpreted as item-categories. Given a user u, we can foresee her preferences on a set of items $\mathcal{I}_u$ by choosing an appropriate user community z (from a set $\{1, \ldots, K\}$) and then choosing an item category w (from a set $\{1, \ldots, L\}$) for each item in the list. The choice of the item category w actually depends on the selected user community z. Finally the preference value is generated by considering the preference of users belonging to the group z on items of the category w. This local modeling of items is the main difference in the generative semantics with respect to state-of-the-art LDA based co-clustering approaches [10].

The generative process for the new *BH-UCM* model, whose corresponding graphical scheme is shown in Fig. 1, is as follows:

- $\forall u \in \mathcal{U}$ sample user community-mixture components $\boldsymbol{\vartheta}_u \sim \mathit{Dirichlet}(\boldsymbol{\alpha})$;

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

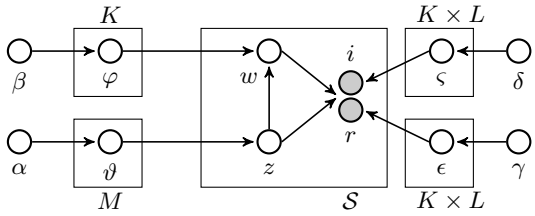


Fig. 1. *BH-UCM* Model

- $\forall z \in \{1, \ldots, K\}$ sample the mixture components $\boldsymbol{\varphi}_z \sim Dirichlet(\boldsymbol{\beta})$
- $\forall w \in \{1, \ldots, L\}, z \in \{1, \cdots, K\}$,
  - Sample item selection components $\boldsymbol{\varsigma}_{z,w} \sim Dirichlet(\boldsymbol{\delta})$
  - Sample rating probabilities $\boldsymbol{\epsilon}_{z,w} \sim Dirichlet(\boldsymbol{\gamma})$
- $\forall u \in \mathcal{U}$
  - Sample the number of items for the user u, $N_u \propto Poisson(\mathcal{K})$
  - For $n = 1$ to N_u
    - Choose a user attitude $z_{u,n} \sim Multinomial(\boldsymbol{\vartheta}_u)$
    - Choose a topic $w_{u,n} \sim Multinomial(\boldsymbol{\varphi}_{z_{u,n}})$
    - Choose an item $i_n \sim Multinomial(\boldsymbol{\varsigma}_{z_{u,n},w_{u,n}})$
    - Generate a rating value for the chosen item according to the distribution $P(r|\boldsymbol{\epsilon}_{z_{u,n},w_{u,n}})$.

Where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ and $\mathcal{K}$ are the fixed hyper-parameters of the model.

Unfortunately, the exact inference for this model is however intractable; hence, we propose a Gibbs Sampling parameter estimation, in which at each step inference can be accomplished by exploiting the full conditional $P(z_n = k_n, w_n = l_n | Z_{\neg n}, W_{\neg n}, \mathbf{R}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. In the latter, z_n (resp. w_n) is the cell of a matrix Z (resp. W) which corresponds to this observation, and $Z_{\neg n}$ (resp. $W_{\neg n}$) denotes the remaining topic assignments. For the n-th observation we have:

$$P(z_n = k_n, w_n = l_n | Z_{\neg n}, W_{\neg n}, \mathbf{R}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \propto$$
$$\frac{n_u^{k_n} + \alpha_{k_n} - 1}{\sum_{k'=1}^{K}(n_u^{k'} + \alpha_{k'}) - 1} \cdot \frac{n_{k_n,i}^{l_n} + \beta_{l_n} - 1}{\sum_{l'=1}^{L}(n_{k_n,i}^{l'} + \beta_{l'}) - 1} \cdot$$
$$\frac{n_{r_n}^{k_n,l_n} + \gamma_{r_n} - 1}{\sum_{r=1}^{V}(n_r^{k_n,l_n} + \gamma_r) - 1} \cdot \frac{n_{i_n}^{k_n,l_n} + \delta_{i_n} - 1}{\sum_{i=1}^{N}(n_i^{k_n,l_n} + \delta_i) - 1}$$

The notation used in the Gibbs Sampling derivation is summarized in Tab. 1. Given the state of the Markov chain, denoted my $\mathcal{M} = (\mathbf{R}, Z, W)$, we can obtain the multinomial parameters $\boldsymbol{\varphi}$, $\boldsymbol{\vartheta}$, $\boldsymbol{\epsilon}$ and $\boldsymbol{\varsigma}$ noticing that, by applying Bayes's rule and then by algebraic manipulations and the properties of the Dirichlet distribution. This ultimately yields the following estimations:

$$\vartheta_{u,k} = \frac{n_u^k + \alpha_k}{n_u + \sum_{k=1}^{K}\alpha_k} \qquad \varphi_{k,l} = \frac{n_{k,l} + \beta_l}{n_k + \sum_{l=1}^{L}\beta_l}$$
$$\epsilon_{k,l,r} = \frac{n_r^{k,l} + \gamma_r}{n_{k,l} + \sum_{r'=1}^{V}\gamma_{r'}} \qquad \varsigma_{k,l,i} = \frac{n_i^{k,l} + \delta_i}{n_{k,l} + \sum_{i'=1}^{N}\delta_{i'}}$$

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

SYMBOL	DESCRIPTION
K	# topics/user communities
L	# item categories
n_u^k	# evaluation of the user u which have been assigned to the user topic k
$n_r^{k,l}$	# times that the rating r has been assigned to each observation when the user topic is k and the item category is l
$n_i^{k,l}$	# times that the item category l has been assigned to observations of the item i when the user topic is k
n_u	# observations for the user u ($\|\mathcal{I}(u)\|$)
n_k	# observations associated with community k
$n_{k,l}$	# times that the category l has been assigned to observations whose user topic is k

Table 1. Summary of notation

Finally, given the pair $\langle u, i\rangle$ we compute the probability of observing the rating value r in a *free prediction* and *forced prediction* context:

$$p(R=r,i|u)=\sum_{k=1}^{K}\sum_{l=1}^{L}\vartheta_{u,k}\cdot\varphi_{k,l}\cdot\epsilon_{k,l,r}\cdot\varsigma_{k,l,i};\qquad p(R=r|u,i)=\frac{p(R=r,i|u)}{p(i|u)}$$

4 Evaluation

For the experimental evaluation of the proposed model, we use two reference benchmark data sets, namely MovieLens-1M[1] and a sample of Netflix data. Both datasets contain explicit preference data: ratings fall within the range 1 to 5, where the latter denotes the highest preference value.
We compare our model with some state-of-the-art competitors from the Collaborative Filtering literature: (PMF) [12], UCM, HUCM [3], and BUCM [2]; and in particular with a selection of co-clustering approaches: *LDCC* [13], *Bregman-CC* [7] and *Bi-LDA* [10].

All models have been trained by retaining the 1% of the training data as held out to perform *early stopping* and avoid overfitting.

Predictive Accuracy. We start our analysis from the evaluation of the prediction accuracy achieved by the algorithms. Table 2 summarizes the best RMSE obtained on both the considered datasets, together with the associated settings.

BH-UCM outperforms all the other co-clustering approaches on the NetFlix data, and is the runner-up winner after *HUCM* which, however, exhibits a marginal advantage. Minimal differences can also be noticed on MovieLens, where *PMF* (a non co-clustering Probabilistic Matrix Factorization approach) achieves the best RMSE score (as expected).

Recommendation Accuracy. Things change substantially when considering the quality of the recommendation in terms of the precision and recall. Based on the

[1] `http://www.grouplens.org/system/files/million-ml-data.tar\_\_0.gz`

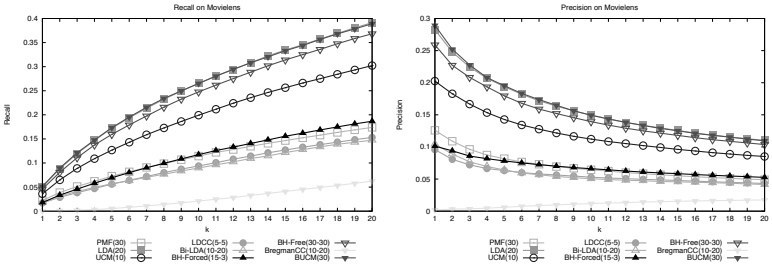
Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

	MovieLens		Netflix	
Approach	**Best RMSE**	**#Topics**	**Best RMSE**	**#Topics**
PMF	**0.8655**	10	0.9309	100
HUCM	0.9278	2-3	**0.9212**	50-10
Bregman-CC	0.9023	10-20	0.9873	3-5
Bi-LDA	0.9033	30-20	0.9362	30-15
LDCC	0.9074	5-5	0.9419	5-10
BH-UCM	0.9073	30-5	0.9256	30-5
BUCM	0.9292	30	0.9431	10

Table 2. Summary of predictive accuracy over the MovieLens and Netflix datasets

results in [1, 2], we consider here also *LDA* model [4], which has been identified as one of the top-performers from this perspective. Notice that *LDA* was not included in the analysis of predictive accuracy, as it does not explicitly support a way to compute rating prediction.

Figure 2 and shows the results of recommendation accuracy on Movielens and Netflix data, when the size k of the recommendation list varies from 1 to 20. We denote by *BH-Free* and *BH-Forced* two different instantiations of the proposed approach, which focus respectively on free and forced prediction. On Movielens data, *BH-UCM* exhibits a minimal worsening on recall (0.39 vs 0.37) and precision (0.11 vs 0.10), against *LDA* and *BUCM*. Notably, the discrepancy between the recommendation accuracy of Bayesian and non-Bayesian approaches is consistently large. In particular, both *BUCM* and *BH-UCM* outperform *UCM*.



(a) Recall on MovieLens (b) Precision on MovieLens

(c) Recall on NetFlix (d) Precision on NetFlix

Fig. 2. Recommendation Accuracy

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

The outperformance of *BUCM* over *BH-UCM* in Movielens can be explained by the different distribution of these data with respect to Netflix. In this latter case, in fact, the huge volume of data is more likely to exhibit local patterns, which are better modeled by *BH-UCM*. By converse, Movielens exhibits both less users and less ratings, and hence the simpler *BUCM* model can easily fit the data.

5 Conclusion

In this work we proposed a hierarchical Bayesian approach for preference data, which extends state-of-the-art (hierarchical) co-clustering techniques, by modeling dynamic associations and dependencies between user- and item-clusters. An extensive evaluation was performed to assess the skills of the devised model, in terms of both rating prediction and recommendation accuracy, showing that the proposed model is competitive with state-of-the-art approaches w.r.t the studied datasets.

References

1. N. Barbieri and G. Manco, *An Analysis of Probabilistic Methods for Top-N Recommendation in Collaborative Filtering*, in Proc. ECML-PKDD Conf., Athens, Greece, 2011.
2. N. Barbieri, G. Costa, G. Manco and R. Ortale, *Modeling Item Selection and Relevance for Accurate Recommendations: A Bayesian Approach*, in Proc. ACM Conf on Recommendation Systems (RecSys'11), 2011.
3. N. Barbieri, G. Manco and E. Ritacco *A Probabilistic Hierarchical Approach for Pattern Discovery in Collaborative Filtering Data*, in Proc. SDM Conf. , 2011.
4. D. M. Blei, A. Y. Ng and M. I. Jordan, *Latent Dirichlet Allocation*, The Journal of Machine Learning Research, 3 (2003), pp. 993–1022.
5. E. Campochiaro, R. Casatta,P. Cremonesi and R. Turrin *Do Metrics Make Recommender Algorithms?*, in AINA Workshops, 2009.
6. P. Cremonesi, Y. Koren and R. Turrin, *Performance of recommender algorithms on top-n recommendation tasks*, in Proc. ACM Conf on Recommendation Systems (RecSys'10), 2010.
7. T. George and S. Merugu, *A Scalable Collaborative Filtering Framework Based on Co-Clustering*, in ICDM, 2005.
8. T. Hofmann, *Latent semantic models for collaborative filtering*, ACM Transactions on Information Systems (TOIS), 22 (2004), pp. 89–115.
9. S. M. McNee, J. Riedl and J. A. Konstan, *Being Accurate is Not Enough: How Accuracy Metrics have hurt Recommender Systems*, in ACM SIGCHI Conference on Human Factors in Computing Systems, 2006.
10. I. Porteous, E. Bart and M. Welling, *Multi-HDP: a non parametric Bayesian model for tensor factorization*, in AAAI, 2008.
11. F. Ricci, L. Rokach,B. Shapira and P.B. Kantor *Recommender Systems Handbook*, Springer, 2011
12. R. Salakhutdinov and A. Mnih, *Probabilistic Matrix Factorization*, in NIPS, 2008.
13. P. Wang, C. Domeniconi and K. B. Laskey, *Latent Dirichlet Bayesian Co-Clustering*, in ECML-PKDD, 2009.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Integrating Clustering Techniques and OLAP Methodologies: The ClustCube Approach*

Alfredo Cuzzocrea[1] and Paolo Serafino[2]

[1] ICAR-CNR and University of Calabria
cuzzocrea@si.deis.unical.it
[2] DIES Dept., University of Calabria
pserafino@deis.unical.it

Abstract. In this paper, we introduce ClustCube, *an innovative OLAP-based framework for clustering and mining complex database objects* extracted from *distributed database settings*. To this end, ClustCube puts together conventional *clustering techniques* and well-consolidated *OLAP methodologies* in order to achieve higher expressive power and mining effectiveness over traditional methodologies for mining tuple-oriented information.

1 Introduction

While lot of proposals on *mining traditional datasets* exist (e.g., [12,3,6,8,2,13,9,7,4]), Data Mining researchers have devoted poor attention to the problem of *effectively and efficiently mining complex objects*, for instance extracted from *distributed database settings* [12]. Contrary to this actual trend, mining complex objects is indeed relevant in practical application scenarios, as *modern database systems are more and more immersed in object-oriented scenarios rather than tuple-oriented scenarios*. Among the wide family of Data Mining techniques available in the active literature, since objects essentially aggregate *low-level fields* (which, in turn, are extracted from attribute values of the underlying distributed database) into *complex classes*, it is natural to think of *clustering* (e.g., [6]) as the most suitable technique to mine such so-derived structures. Also, the combined action of clustering techniques and well-consolidated methodologies developed in the context of *OnLine Analytical Processing* (OLAP) [3] clearly offers powerful tools to mine *clustered objects* according to a multidimensional and multi-resolution vision of the underlying *object domain* [5].

Inspired by these motivations, in this paper we propose *an innovative OLAP-based framework for clustering and mining complex database objects* extracted from *distributed database settings*, called ClustCube, which encompasses a number of research innovations beyond the capabilities of actual Data Mining methodologies over large and complex-in-nature databases (e.g., [12]). To this end, ClustCube combines the power of clustering techniques over complex database objects and the power of OLAP in supporting multidimensional analysis and knowledge fruition of (clustered) complex database objects, with mining opportunities and expressive power infeasible for traditional methodologies. So-obtained ClustCube cubes store clustered complex database objects within cube cells,

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

rather than conventional SQL-based aggregations like in standard *Business-Intelligence*-oriented OLAP data cubes.

Figure 1 shows the "big picture" of the research we propose, i.e. the ClustCube overview. Basically, ClustCube defines a multiple-layer reference architecture that encompasses the following well-separated layers: (*i*) *Distributed DataBase Layer* (DDBL), where the target distributed database from which complex objects are extracted is located; (*ii*) *Complex Object Definition Layer* (CODL), which supports primitives and functionalities for building and managing complex objects extracted from the DDBL layer; (*iii*) the *Object Layer* (OL), where complex objects are located, along with a suitable *object schema*; (*iv*) the ClusterCube *Definition and Management Layer* (CCDML), which supports primitives and functionalities for defining and managing ClustCube cubes; (*v*) the ClusterCube *Layer* (CCL), which stores the final ClustCube cube.

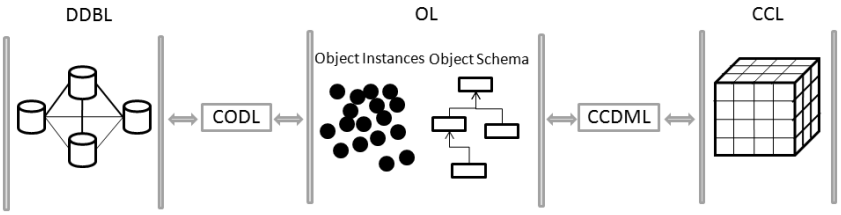


Figure 1. ClustCube Overview

While the mining capabilities exposed by the ClustCube framework are clear and evident enough, a major research challenge in the context of the ClustCube proposal is represented by the issue of efficiently computing ClustCube cubes, enriched by the respective *cuboid lattices* [3], which, similarly to conventional OLAP data cube computation axioms (e.g., [1]) can be extremely resource-consuming on wide collections of objects extracted from large distributed databases. In order to face-off this drawback, we propose algorithms that meaningfully exploit the *structured nature* of complex database objects within cuboids and the *distributive nature* of clustering across hierarchical domains, like those defined by conventional OLAP schemas.

2 Modeling ClustCube Data Cubes

In this Section we provide details on the *ClustCube data cube model*, which is directly inspired from the traditional OLAP data cube model [3], and we describe the solution for computing ClustCube cubes. As mentioned in Section 1, ClustCube cubes store clustered complex objects within cube cells. Complex objects at the OL layer (see Figure 1) are clustered by the CCDML layer on the basis of analysis/mining tasks defined by the administrator and implemented by a suitable class OS_{CODL}, following an innovative clustering approach we describe next. The result of this phase is represented by the materialized ClustCube cube C, which is finally stored in the CCL layer (see Figure 1).

To cluster complex objects and store them within the target ClustCube cube, the CCDML layer makes use of an input clustering algorithm $\mathcal{A}$, which is *orthogonal* to the ClustCube framework itself. This means that any arbitrary clustering algorithm can be exploited within the core of ClustCube, depending on particular characteristics of original data stored in the DDBL layer and the specific clustering goals. In the ClustCube

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

framework, we name $\mathcal{A}$ as *abstract core clustering algorithm* (hereinafter referred as *core algorithm*). At a pure conceptual level, $\mathcal{A}$ captures general concepts that are applicable to every clustering algorithm. In particular, clustering algorithms typically employ a *distance function* [6] in order to determine how objects are partitioned into clusters, and the general goal consists in obtaining clusters having minimal *intra-cluster distance*, i.e. the distance between two objects belonging to the same cluster, and maximal *inter-cluster distance*, i.e. the distance between two objects belonging to different clusters. In our framework, we specialize this classical construct to the case of *distance function over objects*, which relies on the structured nature of objects that, in turn, is based on low-level fields (see Section 1). Obviously, this approach implicitly assumes that *object space is a metric space*, as we demonstrate next.

Given two objects o_i and o_j of class OS_{CODL}, we introduce the following distance function between o_i and o_j $d_{CODL}(o_i, o_j) : OS_{CODL} \times OS_{CODL} \rightarrow \mathbb{R}_0^+$, which is defined as follows:

$$d_{CODL}(o_i, o_j) = \delta_{CODL}(\langle o_i.a_0, \ldots, o_i.a_{|OS_{CODL}|-1}\rangle, \langle o_j.a_0, \ldots, o_j.a_{|OS_{CODL}|-1}\rangle) \quad (2)$$

such that: (*i*) $o_i.a_h$ denotes the h-th field of the object o_i (which, in turn, has been originally extracted from the corresponding attribute value A_h stored in a certain database of the DDBL layer – see Section 1); (*ii*) $\delta_{CODL}(o_i, o_j) : \mathcal{F}(OS_{CODL}) \times \mathcal{F}(OS_{CODL}) \rightarrow \mathbb{R}_0^+$ is a distance function over the metric space induced by the set of fields of the class OS_{CODL}, denoted by $\mathcal{F}(OS_{CODL})$. This metric space is the one taken as reference for the domain of objects of class OS_{CODL}, as mentioned above. It is important to notice here that *the set of* $|OS_{CODL}|$ *fields of class* OS_{CODL} *are used to simultaneously cluster objects of set* OI_{CODL} by means of algorithm $\mathcal{A}$, based on the introduced distance function d_{CODL}. Therefore, OS_{CODL} fields properly play the role of *clustering features* [6]. We formally denote as $\mathcal{F}(OS_{CODL})$ the set of such features. As regards practical issues, it should be noted that clustering algorithm $\mathcal{A}$ analyzes just a sub-set of L features in $\mathcal{F}(OS_{CODL})$, such that $L < |OS_{CODL}|$, in order to cluster objects of class OS_{CODL}, according to well-understood clustering principles [6].

Now, focus the attention on the structure of ClustCube cubes in a greater detail. Given a ClustCube cube C characterized by the *set of dimensions* $\mathcal{D} = \{d_0, d_1, \ldots, d_{N-1}\}$, such that $N = |OS_{CODL}|$, being dimensions in $\mathcal{D}$ corresponding to features in $\mathcal{F}(OS_{CODL})$, each ClustCube cube cell $C[i_0][i_1]\ldots[i_{N-1}] \equiv C[\mathbb{I}]$ in C, such that $0 \leq i_0 \leq |d_0|, 0 \leq i_1 \leq |d_1|, \ldots, 0 \leq i_{N-1} \leq |d_{N-1}|$, denoting $\mathbb{I} = \langle i_0, i_1, \ldots, i_{N-1}\rangle$ an *N-dimensional entry* in the *N-dimensional space* of C, $C[\mathbb{I}]$ stores a set of clustered objects, denoted by $OI_{CODL}(C[\mathbb{I}])$, that *are obtained by simultaneously clustering objects in* OI_{CODL} *with respect to the dimensions/features* $d_0, d_1, \ldots, d_{N-1}$ *in* $\mathcal{D}/\mathcal{F}(OS_{CODL})$. Hence, it is trivial to observe that, for each ClustCube cube cell $C[\mathbb{I}]$ in C *multidimensional boundaries* of $C[\mathbb{I}]$ along the dimension d_i, denoted by $B_{d_i}^{low}$ and $B_{d_i}^{up}$, respectively, with $B_{d_i}^{low} < B_{d_i}^{up}$, are determined by the $\mathcal{A}$-based clustering of objects in OI_{CODL} with respect to feature d_i in $\mathcal{D}$. In other words, while in traditional OLAP data cubes [3] the multidimensional boundaries of data cube cells along dimensions are determined by the input *OLAP aggregation scheme* (e.g., [1]), in ClustCube cubes multidimensional boundaries of ClustCube cube cells along dimensions are the *result* of the clustering process itself.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

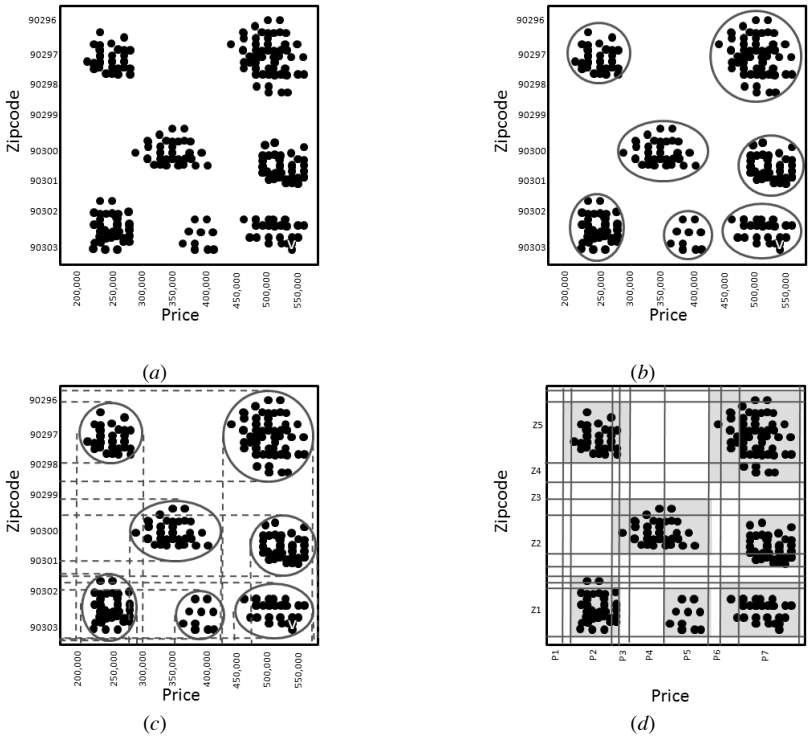


Figure 2. Traditional OLAP Scheme-Driven Aggregation (*a*) and ClustCube Clustering-Guided Aggregation (*b*) for an Example Two-Dimensional House Sale Cube

This novelty has deep consequences even in the way cube aggregations are computed. To become convinced of this, consider a simple case study focused on an house sale dataset that logically defines a two-dimensional space characterized by the following dimensions (features, respectively – see Figure 2 (*a*)): (*i*) *Price*, which represents the price at which a certain house is sold; (*ii*) *Zipcode*, which represents the zipcode of the city where the house is located. Figure 2 (*b*) shows a possible clustering of such dataset with respect to the features (dimensions, respectively) *Price* and *Zipcode*. Figure 2 (*c*) shows instead the *projection* of so-obtained clusters along both the dimensions, and, finally, Figure 2 (*d*) shows the final (logical-representation) of the two-dimensional ClustCube cube, where blue lines denote cube cells boundaries. It should be noted that, contrary to traditional OLAP data cubes where tuples are aggregated according to *regular* and *somewhat natural* groups along dimensional hierarchies defined by the input OLAP aggregation scheme, thus determining *regular partitions* of the target data domain, in novel ClustCube data cubes groups of objects stored in (ClustCube) cube cells correspond to clusters computed by input clustering algorithm $\mathcal{A}$, thus determining irregular partitions of the target object domain following sort of a *clustering-guided ClustCube aggregation scheme*. As shown in Figure 2 (*d*), given a ClustCube data cube C, every cell $C[i_0][i_1]\ldots[i_{N-1}]$ in C may alternatively contain either a whole cluster generated from $\mathcal{A}$ or a sub-cluster of it.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

3 Computing ClustCube Data Cubes

In this Section we provide details on how to *compute ClustCube data cubes*. One of the most relevant contribution of our research consists in equipping the final ClustCube cube generated by the CCDML layer (see Figure 1) with the canonical cubod lattice [3]. In traditional OLAP, given an N-dimensional data cube C having $\mathcal{D} = \{d_0, d_1, \ldots, d_{N-1}\}$ as dimension set, the cuboid lattice associated to C, denoted by $\mathcal{L}$, is a *hierarchical structure* composed by $2^N - 1$ cuboids, denoted by C_i, i.e. data (sub-)cubes that aggregate original relational data according to arbitrary combinations of dimensions in $\mathcal{D}$, each one at different cardinality. In other words, an n-dimensional cuboid C_i of a data cube C represents a particular n-dimensional view of C, such that $0 \leq n \leq N$. For $n = N$, the base cuboid is defined, which corresponds to the original data cube C. Cuboids of a certain cuboid lattice $\mathcal{L}$ are naturally ordered by means of the *precedence relation* $\prec$, such that, for each pair of cuboids C_i and C_j in $\mathcal{L}$, $C_i \prec C_j$ holds iff $\mathcal{D}_i \subset \mathcal{D}_j$, such that $\mathcal{D}_i$ denotes the set of dimensions of C_i and $\mathcal{D}_j$ denotes the set of dimensions of C_j, respectively. This finally determines a *cuboid hierarchy*, denoted by $\mathcal{H}(\mathcal{L})$. For instance, Figure 3 shows in the left side the cuboid lattice $\mathcal{L}$ for a four-dimensional ClustCube cube C having $\mathcal{D} = \{A, B, C, D\}$ as dimension set. Here, for instance, the property $CD \prec BCD$ holds in $\mathcal{H}(\mathcal{L})$. Also, from Figure 3 (left side), it should be clear enough that in the cuboid lattice $\mathcal{L}$ of an N-dimensional data cube C, cuboids are structurally organized according to $N + 1$ levels such that l-dimensional cuboids are located at level l of $\mathcal{L}$ are hierarchically linked to cuboids at level $l - 1$ and $l + 1$ of $\mathcal{L}$, respectively.

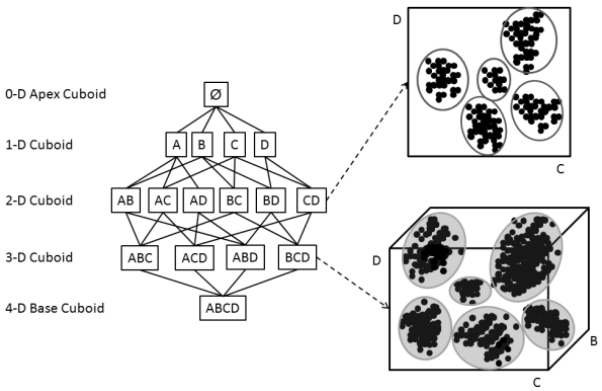


Figure 3. Cuboid Lattice of an Example Four-Dimensional ClustCube Cube, with Details on Cuboids *CD* and *BCD*

Now, focus the attention on some important properties of the ClustCube data cube model with respect to the proper clustering. Figure 3 shows in the right side clustered objects stored by the cuboids *CD* and *BCD*, respectively. Here, it should be clear enough that cuboid *BCD* stores clusters (of objects) that are *conceptually* obtained from clusters of cuboid *CD* (which *precedes BCD* in $\mathcal{H}(\mathcal{L})$, i.e. $CD \prec BCD$) *by distributing objects in CD with respect to the newly-added dimension/feature B*. In the ClustCube framework, we

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

fully take advantages from such a distributed nature of clustering across hierarchical cuboids, being this nice amenity the key property that allows us to significantly reduce computational needs due to compute ClustCube cube cuboid lattices. In fact, thanks to this amenity, *we do not need to compute cuboids from the scratch* but any cuboid C_j at level l of $\mathcal{L}$, such that $2 \leq l \leq N$, can be obtained from the cuboids at level $l = 1$, denoted by $\{C_0, C_1, \ldots, C_{N-1}\}$, simply by *simultaneously distributing* objects in C_i with respect to the features $\{\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{N-1}\}$ of $\{C_0, C_1, \ldots, C_{N-1}\}$, respectively.

Given the input class OS_{CODL}, the collection of complex objects OI_{CODL} and the input clustering algorithm $\mathcal{A}$, the CCDML layer computes the final ClustCube cube to be stored at the CCL layer (see Figure 1). To this end, ClustCube framework comprises several kinds of techniques for efficiently building the ClustCube cube C plus its cuboid lattice $\mathcal{L}$ (each one codified by a respective ClustCube cube building algorithm), which differ with respect to two *orthogonal* strategies, namely: (*i*) *materialization strategy* and (*ii*) *building strategy*. Materialization strategies specify which cuboids, among the $2^N - 1$ cuboids of $\mathcal{L}$, must be *materialized*, i.e. *computed* and *stored* (in secondary memory). On the other hand, building strategies specify how cuboids are *computed* actually. With respect to the materialization strategy, the following two alternatives are introduced in the ClustCube framework: (*i*) *full*, denoted by FUL, according to which *all* cuboids of $\mathcal{L}$ are materialized; (*ii*) *partial*, denoted by PAR, according to which a *sub-set* of the $2^N - 1$ cuboids of $\mathcal{L}$ is materialized. As regards the building strategy, ClustCube framework exposes the following two different approaches: (*i*) *baseline*, denoted by BAS, according to which, for each cuboid C_i in $\mathcal{L}$, clusters are *re-computed from the scratch* (i.e., directly from input objects in OI_{CODL}); (*ii*) *drill-down*, denoted by DRI, according to which cuboids at level l of $\mathcal{L}$ are computed from cuboids at level $l = 1$ of $\mathcal{L}$ by means of a meaningfully distributive method. It is critical to notice here that, in the ClustCube framework, the target ClustCube cube C is obtained via computing the whole cuboid lattice $\mathcal{L}$ in terms of the base cuboid [3].

4 Conclusions and Future Work

A complete OLAP-based framework for clustering and mining complex objects extracted from distributed database settings, called ClustCube, has been presented in this paper. ClustCube encompasses a spectrum of research innovations towards the seamless integration of consolidated clustering techniques over large databases and well-understood OLAP methodologies for accessing and mining (complex) objects according to a multidimensional and multi-resolution vision of the underlying object domain. Future work is mainly oriented towards extending the proposed framework in order to make it able of dealing with *classification issues over complex database objects*, beyond clustering issues like those investigated in this research.

References

1. Agarwal, S., Agrawal, R., Deshpande, P., Gupta, A., Naughton, J.F., Ramakrishnan, R., Sarawagi, S.: On the Computation of Multidimensional Aggregates. In: *Proc. of VLDB 1996*, pp. 506—521 (1996)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *Proc. of KDD 1996*, pp. 226—231 (1996)
3. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub Totals. *Data Mining and Knowledge Discovery 1(1)*, pp. 29—53 (1997)
4. Han J.: Towards On-line Analytical Mining in Large Databases. *ACM SIGMOD Record 27(1)*, pp. 97—107 (1998)
5. Han, J., Kamber, M.: *Data Mining: Concepts and Techniques, second ed.* Morgan Kauffmann Publishers, San Francisco, CA, USA (2006)
6. Hinneburg, A., Keim, D.A.: Clustering Methods for Large Databases: From the Past to the Future. In: *Proc. of ACM SIGMOD 1999*, p. 509 (1999)
7. Kriegel, H.-P., Kröger, P., Zimek, A.: Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. *ACM Transactions on Knowledge Discovery from Data 3(1)*, pp. 1—58 (2009)
8. Ng, R.T., Han, J.: CLARANS: A Method for Clustering Objects for Spatial Data Mining. *IEEE Transactions on Knowledge and Data Engineering 14(5)*, pp. 1003—1016 (2002)
9. Sheikholeslami, G., Chatterjee, S., Zhang, A.: WaveCluster: A Wavelet Based Clustering Approach for Spatial Data in Very Large Databases. *VLDB Journal 8(3-4)*, pp. 289—304 (2000)
10. Transaction Processing Council, *TPC Benchmark H*, available at http://www.tpc.org/tpch/
11. Xin, D., Han, J., Xiaolei, L., Shao, Z., and Wah, B.W.: Computing Iceberg Cubes by Top-Down and Bottom-Up Integration: The StarCubing Approach. *IEEE Transactions on Knowledge and Data Engineering 19(1)*, pp. 111—126 (2007)
12. Yin, X., Han, J., Yu, P.S.: CrossClus: User-Guided Multi-Relational Clustering. *Data Mining and Knowledge Discovery 15(3)*, pp. 321—348 (2007)
13. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: A New Data Clustering Algorithm and Its Applications. *Data Mining and Knowledge Discovery 1(2)*, pp. 141—182 (1997)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Enhancing Datalog with Epistemic Operators to Reason About Knowledge in Distributed Systems*

Matteo Interlandi

University of Modena and Reggio Emilia
matteo.interlandi@unimore.it

Abstract. In the last few years, researchers started to investigate how recursive queries and deductive languages can be applied to find solutions to the new emerging trends in distributed computing. We conjecture that a missing piece in the current state-of-the-art in logic programming is the capability to express statements about the knowledge state of distributed nodes. In fact, reasoning about the state of remote nodes is fundamental in distributed contexts in order to design and analyze protocol behavior. To reach this goal, we leveraged Datalog$^\neg$ with an epistemic modal operator, allowing the programmer to directly express nodes' state of knowledge instead of low level communication details. To support the effectiveness of our proposal, we introduce, as example, the declarative implementation of a well-known protocol employed to execute distributed databases transactions: the two phase commit protocol.

1 Introduction

Pushed by the new interest that Datalog is acquiring in the database community, the goal of this paper is to open a new direction in the investigation on how Datalog could be adopted to program distributed systems. Many authors have stated how logic programming in general [6] and Datalog in particular [5] seems to particularly fit the representation of distributed programs implementation and properties. We think that a missing point is the possibility to express statements about the knowledge state of distributed nodes in Datalog. In fact, the ability to reason about the knowledge state of remote nodes has been demonstrated [4] to be a fundamental tool in multi-agent systems in order to specify global behaviors and properties of protocols. Motivated by all these facts, we leveraged Datalog$^\neg$ with an epistemic modal operator, allowing the programmer to express directly nodes' state of knowledge instead of low level communication details. To support our assertions, we describe our implementation of the two phase commit protocol. The remainder of the paper is organized as follow: Section 2 contains some preliminary notations about Datalog$^\neg$. Section 3 describes what we intend for

* This work is partially taken from [9].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

a distributed system and introduces some concept such as *global state*, *run* and the modal operator K. Section 4 introduces Knowlog and the implementation of the two phase commit protocol. The paper finish with conclusions and future work.

2 Preliminaries

In order to define Knowlog, we first introduce some principles of Datalog$^\neg$ [1], and Datalog$^\neg$ augmented with temporal constructs [8, 3]. A Datalog$^\neg$ rule is an expression in the form:

$$H(\bar{u}) \leftarrow B_1(\bar{u}_1), ..., B_n(\bar{u}_n), \neg C_1(\bar{v}_1), ..., \neg C_m(\bar{v}_m)$$

where $n, m \geq 0$, H, B_i, C_j are relation names $i = 0, ..., n$ and $j = 0, ..., m$ and $\bar{u}, \bar{u}_i, \bar{v}_j$ are tuples of appropriate arities. Tuples are composed by *terms* and each term can be a constant in the domain **dom** or a variable in the set **var**. We will use interchangeably terms *predicates* and *relations*. As usual $H(\bar{u})$ is referred as the *head*, $B_i(\bar{u}_i)$, $C_j(\bar{v}_j)$ as the *body*, and in general $H(\bar{u})$, $B_i(\bar{u}_i)$ and $C_j(\bar{v}_j)$ as *atoms*. A *literal* is an atom (in this case we refer to it as *positive*) or the negation of an atom. If $m = n = 0$ and does not contains variable terms, the rule express a *fact* or equivalently a *groud atom*. In this paper we assume that each rule is *range restricted*, i.e. every variable occurring in a rule-head appears in at least one positive literal of the rule body. Then, a *Datalog$^\neg$ program* Π is a set of range restricted rules. For a database schema **R**, a *database instance* is a finite set **I** constructed by the union of the relation instances over R with $R \in \mathbf{R}$ a relation name and where each relation instance is a finite set of facts. As introductory example, we use the program depicted in Listing 1.1 where we used a relation `link`, containing tuples in the form `(S,D)`, to specify the existence of a link between a source node `S` and a destination node `D`. In addition we employ the `path` relation, which is computed starting from the `link` relation (`r1`) and recursively adding a new path when, roughly speaking, there is a link from `A` to `B` and already exists a path from `B` to `C` (`r2`).

```
r1: path(X,Y):-link(X,Y)
r2: path(X,Z):-link(X,Y),path(Y,Z)
```

Listing 1.1. Simple Recursive Datalog Program

2.1 Time in Datalog$^\neg$

With the language we are introducing, we want to model programs for distributed systems. These systems are not static, but evolving with time. Therefore it will be useful to enrich Datalog$^\neg$ with some notion of time. To reach this goal we follow the road traced by Statelog [8] and Dedalus [3]. Thus, informally, each relation is labeled with a *time-step identifier* having values in $\mathbb{N}$ and which specify at what time-step a given instance has been derived and is true. A consequence of this approach is that tuples by default are considers *ephemeral*, i.e., they are valid only for one single time-step. Obviously, tuples can became *persistent* - once derived, for example at time s, they last for every time $t \geq s$ - if

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

they are stored in *persistent* relations. Among the different temporal extensions of Datalog¬ available in the literature, we embrace the Dedalus [3] notation, thus programs' rules are divided in two sets: *inductive* and *deductive*. The former set contains all the rules employed for transfer tuples among time-steps i.e., persistency rule, while the latter encompasses the rules that are local into a single time-step. Some syntactic sugar is adopted in order to better characterize rules and relations: deductive rules appears as usual Datalog¬ rules, while a `next` suffix is introduced in head relations to characterize inductive rules. In Listing 1.2 the simple program of the previous section is rewritten to introduce the new formalism: a persistent relation rule (`r1`) and a rule for the modification of the `link` relation if a tuple is issued to the ephemeral relation `link_down`, representing an event on the link which cause the link to be disconnected.

```
r1: link(X,Y)@next:-link(X,Y),¬del_link(X,Y)
r2: del_link(X,Y):-link_down(X,Y)
r3: path(X,Y):-link(X,Y)
r4: path(X,Z):-link(X,Y),path(Y,Z)
```

Listing 1.2. Inductive and Deductive Rules

3 Distributed Logic Programming

Before starting the discussion on how we leverage the language with epistemic operators, we first introduce our model of distributed system and how communication among nodes is performed. We define a distributed message-passing system to be a non empty finite set N of share-nothing nodes joined by bidirectional communication links. Each node identifier has a value in the domain **dom** but here we consider the set $N = \{1, ..., n\}$ of node identifiers, where n is the total number of nodes in the system. We identify with *adb* a new set of *accessible* relations encompassing all the tables that are horizontally partitioned among nodes and through which nodes are able to communicate. Each relation $R \in adb$ contains a *location specifier* term [7]. This term maintains the identifier of the remote node to which every new fact inserted into the relation R must be issued. As pointed out in [5, 3], modeling communication using relations provides major advantages. Continuing with the examples introduced in the previous sections, in order to describe Listing 1.3 we can imagine a real network configuration where each node has locally installed the program, and where each `link` relation reflect the actual state of the connection between nodes. For instance, we will have the fact `link(A,B)` in node A's instance if a communication link between A and node B exists. The location specifier term is identified by the `@` prefix.

```
r1: link(X,Y)@next:-link(X,Y),¬del_link(X,Y)
r2: del_link(X,Y):-link_down(X,Y)
r3: path(@X,Y):-link(X,Y)
r4: path(@X,Z):-link(X,Y),path(@Y,Z)
```

Listing 1.3. Distributed Program

The semantics of the program in Listing 1.3 is the same as in the previous section, even though operationally it substantially differs. In fact, in this new

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

version, computation is performed simultaneously on multiple distributed nodes. Communication is achieved through rule `r4` which, informally, specify that a path from a node A to a node C exists if there is a link from A to another node B and this last knows that exist a path from B to C.

3.1 The Knowledge Model

In every point in time, each node is in some particular *local state* incapsulating all the information the node possesses. We use s_i to denote the local state of node i . We define the *global state* of a distributed system as a tuple $(s_1, ..., s_n)$ where s_i is the node i's state. We define how global states may change over time through the notion of *run*, which binds time values to global states, i.e., $r : \mathbb{N} \to \mathcal{G}$ where $\mathcal{G} = \{S_1 \times ... \times S_n\}$ and S_i is the set of possible local state for node $i \in N$. Following [4] we define a *system* as a set of runs. Using this definition we are able to deal with a system not as a collection of interacting nodes but, instead, directly modeling its behavior, abstracting away many low level details. In knowledge-based systems, nodes are able to accomplish actions not only based on their local state, but also on the knowledge the node has, i.e., the information the node has about the state of the system. If we consider two runs of a system, with global states respectively $g = (s_1, ..., s_n)$ and $g' = (s'_1, ..., s'_n)$, g and g' are *indistinguishable* for process i, and we will write $g \sim_i g'$ if i has the same local state both in g and g', i.e., $s_i = s'_i$. We use the modal operator K_i and we write $K_i\psi$ to express that a node i knows sentence ψ: in every global state that i considers possible - i.e., all the global state that are indistinguishable for i - the sentence ψ is true. This definition of knowledge follows the axioms that are called *S5*. We refer the reader to [9] for a detailed discussion about the modal operator K.

4 Incorporating Knowledge: Knowlog

We employ $\Box$ to denote a (possible empty) sequence of modal operators K and we use the following statement to express it in a rule form:

$$\Box(H \leftarrow B_1, ..., B_n, \neg C_1, ..., \neg C_m) \tag{1}$$

with $n, m \geq 0$ and each positive literal is in the form $\Box R$, while negative literals are in the form $K_i\Box R$ where K_i is equal to the *model context*. From [10] we adopt the term *modal context* to refer to the sequence - with the maximum length of one - of modal operators appearing in front of a rule. We put some restriction on the sequence of operators permitted in $\Box$.

Definition 1. *Given a (possibly empty) sequence of operators $\Box$, $\Box$ is in restricted form if it does not contain K_iK_i subsequences, with i specifying a process identifier.*

Definition 2. *A Knowlog program is a set of rules in the form (1), containing only (possible empty) sequences of modal operators in the restricted form and where the subscript i of each modal operator K_i can be a constant or a variable.*

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Informally speaking, given a Knowlog program, using modal context we are able to assign to each node the rules the node is responsible for, while atoms and facts residing in the node i are in the form $K_i \Box R$. We define communication rules as follow:

Definition 3. *A communication rule in Knowlog is a rule where no modal context is set and the body atoms have the form $K_i \Box R$ - namely they are prefixed with modal operators related to the same process - while the head atom has the form $K_j \Box R'$, with $i \neq j$ and not necessarily $R' \neq R$.*

In this way, we are able to abstract away all the low level details about how information is exchanged, leaving to the programmer just the task to specify what a process should know, and not how. For the definition of the Knowlog semantics we refer to [9].

The Two-Phase-Commit Protocol Inspired by [2], we implemented the two-phase-commit protocol (2PC) using the epistemic operator K. 2PC is used to execute distributed databases transaction and it is divided in two phases: in the first phase, called the *voting phase*, a coordinator node submit to all the transaction's participants the willingness to perform a distributed commit. Consequently, each participant sends a vote to the coordinator, expressing its intention. In the second phase - namely the *decision phase* - the coordinator collects all votes and decides if performing global *commit* or *abort*. The decision is then issued to the participants which act accordingly. In the 2PC implementation of Listing 1.4, we assume that our system is composed by three nodes: one coordinator and two partecipants. Due to the lack of space, we considerably simplify the 2PC protocol. For a more detailed discussion we refer the reader to [9].

```
\\Initialization at coordinator
   r1: Kc(part_cnt(count<N>):-nodes(N))
   r2: Kc(transaction(Tx_id,State):-log(Tx_id,State))
\\Decision Phase at coordinator
   r3: Kc(yes_cnt(Tx_id,count<part>):-vote(Vote,Tx_id,part),Vote == "yes"))
   r4: Kc(log(Tx_id,"commit")@next:-part_cnt(C),yes_cnt(Tx_id,C1),C==C1,
        State=="vote-req",transaction(Tx_id,State))
   r5: Kc(log(Tx_id,"abort"):-vote(Vote,Tx_id,part),Vote == "no",
        transaction(Tx_id,State),State =="vote-req")
\\Voting Phase at partecipants
   r6: Kp(log(Tx_id,"prepare"):-State=="vote-req",Kctransaction(Tx_id,State))
   r7: Kp(log("abort",Tx_id):-log(Tx_id,State),State=="prepare",
        db_status(Vote),Vote=="no")
\\Decision Phase at partecipants
   r8: Kp(log(Tx_id,"commit"):-log(Tx_id,State_l),State_l=="prepare",
        State_t=="commit",Kctransaction(Tx_id,State_t))
   r9: Kp(log(Tx_id,"abort"):-log(Tx_id,State_l),State_l=="prepare",
        State_t=="abort",Kctransaction(Tx_id,State_t))
\\Communication
   r10:Kxtransaction(Tx_id, State):-Kcsubs(X),
```

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

```
     Kctransaction(Tx_id,State),Kcpath(@Y,X)
r11:Kcvote(Vote,Tx_id,"sub1"):-KP1log(Tx_id,State),
     State=="prepare",KP1db_status(Vote),KP1path(@P1,C)
r12:Kcvote(Vote,Tx_id,"sub2"):-KP2log(Tx_id,State),
     State=="prepare",KP2db_status(Vote),KP2path(@P2,c)
```

Listing 1.4. Two Phase Commit Protocol

In the above example, for simplicity we wrote K_{P} as a modal context instead of K_{P1} and K_{P2}.

5 Conclusion and Future Work

In this paper we present Knowlog, a programming language based on Datalog$^{\neg}$ leveraged with a notion of time and modal operators. Through Knowlog, reasoning about state of knowledge in distributed systems can be performed, therefore lighten the programmer's burden of expressing low level communication details. What we discussed here is a first step towards the definition of a comprehensive logical framework able to define a declarative as well as operational semantics, and generic enough to be adopted in multiple contexts. We are confident that following our approach, properties such as processes coordination and replicas consistency can be exhaustively defined.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Alvaro, P., Condie, T., Conway, N., Hellerstein, J.M., Sears, R.: I do declare: consensus in a logic language. Operating Systems Review **43**(4) (2009) 25–30
3. P. Alvaro, W. R. Marczak, N. Conway, J. M. Hellerstein, D. Maier, and R. Sears. Dedalus: Datalog in time and space. Datalog Reloaded - First International Workshop, Datalog 2010, Oxford, UK, 2010, 262–28.
4. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press, Cambridge, MA, USA (2003)
5. Hellerstein, J.M.: The declarative imperative: experiences and conjectures in distributed logic. SIGMOD Rec. **39** (September 2010) 5–19
6. Lamport, L.: The temporal logic of actions. ACM Trans. Program. Lang. Syst. **16** (May 1994) 872–923
7. Loo, B.T., Condie, T., Garofalakis, M., Gay, et al. : Declarative networking: language, execution and optimization. In: SIGMOD '06, New York, NY, USA, ACM (2006) 97–108
8. Ludäscher, B.: Integration of Active and Deductive Database Rules. Volume 45 of DISDBIS. Infix Verlag, St. Augustin, Germany (1998)
9. Interlandi, M.: Knowlog: A Declarative Language for Reasoning about Knowledge in Distributed Systems. ER'12 PhD Symposium Florence, Italy (2012)
10. Nguyen, L.A.: Foundations of modal deductive databases. Fundam. Inf. **79** (January 2007) 85–135

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

On Casanova and Databases or the Similarity Between Games and DBs*

Giuseppe Maggiore, Renzo Orsini, and Michele Bugliesi

Università Ca' Foscari Venezia
DAIS – Computer Science
{maggiore,orsini,bugliesi}@dais.unive.it

Abstract. In this paper we discuss the similarities between two fields which are traditionally considered worlds apart: game development and databases. We discuss how many aspects of game development either use databases, data-mining, etc. directly to solve challenging data-management problems, but also how the game logic itself is subtly related to many techniques and theoretical results already explored in the field of databases. We also discuss our Casanova language, which is a game development language which we are building with the explicit aim of taking advantage of this relationship, in order to greatly simplify the craft of game-making by introducing automated optimizations and declarative constructs to define a game with less boilerplate code.

Keywords: Game development, Casanova, databases, languages, functional programming

1 Introduction

Games are a growing field, rapidly approaching in size and audience the music and movie industries [1]. Game development goes beyond entertainment: serious games [2] experiment with the use of gameplay to teach important lessons, while [3] even use interactive game development to teach Computer Science to young students. Also, the same tools and techniques used to create games are used for virtual reality and interactive simulations in general. Unfortunately games are very much unexplored territory when it comes to disciplined research, a research that would be much needed to contain the skyrocketing costs involved with creating a modern game. Reducing the costs of game-making would allow researchers and practitioners alike to explore new and interesting games without the risk of a very large initial investment. This may indeed be the reason why a large number of modern commercial games destined to the mass-market are built with so little daring in terms of exploring new gameplay and new mechanics: to reduce the risks of exploring new territories.

Where does the difficulty in making games come from? Game development is so expensive because to ensure that the final result stands up to the user expectations [4] a game needs a high visual quality to clearly show the various states of the logical entities of the game, and the logical entities of the game must be updated according to

* Extended Abstract

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

an articulated simulation that evolves their state in a meaningful way. The visual and logical modules of the game are large and complex to build; to make matters worse, both must run in a loop that is optimized enough to update the screen and animate the game entities in real-time, that is all iterations of the game logic and drawing must be completed in a time span that ranges between $1/20^{th}$ and $1/60^{th}$ of a second. As an additional challenge, game-making comprises a (rather large) creative portion that is performed by designers, who rarely are well-versed in the arcana of computer programming: for this reason the architecture of a game must be flexible and easily modifiable so that designers can quickly build and test new iterations of gameplay. Finally, the surrounding system of a modern multiplayer game poses additional challenges, such as synchronizing the game world across many clients, *always in real-time*, clustering players by skill-level to create balanced matches, and so on. In short, games offer a unique blend of complexity, optimization, and need for customization by non-programmers which yields very high construction and maintenance costs.

In this paper we present a novel observation: many game development problems may be already solved in a field which, at a first glance, may appear utterly unrelated. The field of databases already contains a large body of relevant research works which simply needs to be studied and adopted by game developers. For this purpose we are creating a game development framework, called Casanova [5,6,7], that simplifies game development along this direction. We will start with a discussion about coding the logical simulation of a game in Section 2; we then study the matters of world persistency in a "massively multiplayer online game" (MMOG) in Section 3; finally, we conclude with a remark on mining players data for match-making or preference deduction in Section 4.

1.1 Related work

There is at least one other underway research effort of linking database research with game development; this work has yielded the SGL language [8], an experimental game development language which uses SQL queries to define the way the various entities of the game world are updated at each time-step of the simulation. SGL may be unsuitable for larger scale problems, since it offers no techniques to model the game world and entities, but the underlying optimizations and expressivity of the framework are undeniably very powerful and require virtually no effort on the part of the game developer.

The present work also builds on our database-inspired game development language, Casanova [5] [7] [6]. The language aims at offering a series of abstractions and optimizations that allow a developer to specify only certain core aspects of a game logic and visualization, without concerning himself too much with boilerplate code such as state traversal or query optimizations.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2 The Game World

The logical simulation of a game starts exactly at the first step of the creation of a new database: modeling (or conceptual schema definition [9]). A game consists, at its core, of a series of concepts and their relationships. This describes the semantics of a game world and represents a series of assertions about its nature. Specifically, it describes the things of significance to the game, about which it is inclined to collect information, and characteristics of (*attributes*) and associations between pairs of those entities (*relationships*). Most entities are in the plural, and thus require being stored in tables or collections. For example, the Game of Life might be modeled in Casanova as:

```
type World = { Cells : list<list<Cell>> }
type Cell = {
  NearCells   : list<ref<Cell>>
  Value       : int }
```

After defining the data model of the game world, game developers must define the *dynamics* of the game, that is how each game entity is updated at every tick of the game loop. The game dynamics is, at its core, a series of rules that define how each entity (or, better, each attribute of each entity) is updated during each tick. A major point of difference between games and databases lies in the frequency of the dynamics of the system: the game world is updated about once every sixtieth of a second to achieve a smooth simulation, instead of waiting for user-initiated events; indeed, a large number of changes in the game world are entirely automated and occur naturally over time. In Casanova the game dynamics is computed by stating a series of rules for each field of the game that needs updating, that is the above definition would also include:

```
type Cell = {
  NearCells   : list<ref<Cell>>
  Value       : int }
 rule Value(world,self,dt) =
  let around = sum [c.Value | c <- self.NearCells]
  match around with
  | 3 -> 1
  | 2 -> self.Value
  | _ -> 0
```

Some of the game dynamics simply require to recomputed simple values, while others require more sophistication. Consider a game where we need to compute the collisions between asteroids and projectiles; we might define a projectile so that it computes a query on the entire game world to find the list of asteroids colliding with itself at each tick:

```
type Projectile = {
 Position       : vector2<m>
 Velocity       : vector2<m/s>
 Colliders      : list<ref<Asteroid>> }
 rule Position(world,self,dt) = self.Position + self.Velocity * dt
 rule Colliders(world,self,dt) =
  [x | x <- world.Asteroids && distance(self.Position, x.Position) < 10.0f]
```

Notice that certain attributes of the game entities are marked with the ref data constructor, which represents referential constraints (foreign keys) [10] between different lists of entities; references define attributes which do not contain entities to be updated during a tick; in our example above this means that the colliders of a projectile are

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

not asteroids to be updated during a tick, but just a series of asteroids which we need for certain logical computations to be associated with a certain projectile.

Rules are treated as transactional operations [11] in order to ensure the consistency of the game world. This means that all rules are evaluated on the game world at a certain time-step ($world_t$) and then all their results are written, at the same time, into the new game world ($world_{t+1}$). This way all rules behave in a predictable way and no rule ever "sees" the game world halfway between different ticks of the simulation. Moreover, this enables a very important optimization: evaluating rules in parallel with different threads so as to speed up the simulation, thus freeing computational power to animate more entities or use more complex algorithms.

Rules on collections also present an additional optimization opportunity: certain operations (the colliders example above is a particularly fitting example) need to compute a Cartesian product between two lists, asteroids and projectiles, which naïvely computed would have quadratic complexity. By using optimization techniques such as a hash-join or similar the complexity becomes much lower. Our benchmarks [7] suggest improvements of an order of magnitude in the run-time efficiency of the entire simulation when applying query optimization techniques.

Rules and queries are not always the best abstraction to represent the way a game world evolves itself over time. For this reason we have added to Casanova a scripting system, which is decidedly akin to a system of triggers and stored procedures (where triggers may also be timers or user actions). The (soft) real-time constraint of a game requires that our procedures do not block a game tick for an excessive period of time, because otherwise this would break the smoothness of the user experience. To better mix the game loop and the evaluation of these procedures we have implemented them as coroutines [5], that is they feature a yield statement that suspends procedure evaluation until the next tick, and the wait statement that suspends procedure evaluation for a given amount of time. We would define a Casanova script that waits for the player to press a button to shoot a projectile by writing:

```
{ if is_key_down Keys.Space then return Some() else return None } => {
  world.Projectiles.Add
    { Position = vector(50.0<, 0.)
      Velocity = vector2(cos(state.CannonAngle),sin(state.CannonAngle))
      Colliders = [] }
  wait 0.1<s> }
```

3 Persistency, Saving Games and Multiplayer Games

A game world requires some persistency. Persistency comes into play both in single-player games and multi-player games. Single-player games require persistency because the playing experience takes much longer than a single play session, and so the game state requires serialization on persistent memory. The act of storing and retrieving the game world from persistent memory must be quick, since saving the game can be (and often is) done during gameplay and thus must be optimized for speed as the rest of the game, to avoid breaking the flow of gameplay.

A more complex case where the game world is persistent is that of multiplayer games. Multiplayer games have two different sets of problems to tackle: *(i)* synchronizing the

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

game world in real-time between different clients; and *(ii)* reliably storing a persistent world and all the players' data.

Synchronization of the game world between many clients and the game server (or *host*) must happen in real-time, but each client needs a responsive experience. For this reason most modern games employ client-side prediction and lag-compensation algorithms [12], that is all operations that need to write the host' game world always appear to succeed locally and is then validated (as soon as possible considering the roundtrip time for unreliable networked messages) by the host. This amounts to a form of eventual consistency.

The problem of storing a persistent, huge world for many players (games such as World of Warcraft feature *millions* of concurrent players) requires hybrid in-memory/on-disk databases with very quick access and supporting up to hundreds of thousands of concurrent accesses. To reduce the scope of these technical challenges the game world is sometimes segmented into different copies of the world, grouping players by geographical reason, but other games such as EVE Online feature different techniques such as a hierarchical structure of distributed servers to avoid segmentation and offer a single persistent game world.

4 Match-making, Understanding Players

Another challenge that multiplayer games face is that of making use of the huge amount of data that can be gathered from players' behavior through data-mining techniques. A common instance of this is the match-making problem [13]: given the preferences of the players who are currently waiting to start a game, determine the ideal group of players who all have similar skills, low-latency between each other, similar preferences, etc. Some games even feature team-play, and so the best teams must be defined automatically.

Similarly, game developers often need to understand the preferences of their players or if there are certain conditions in the game that favor certain players, in order to maintain a fun, balanced and fair experience for everyone. Discriminating useful patterns from previous games logs requires the ability to wade through huge data bases to make sense of their information.

5 Conclusions and Future Work

Game development is a large and important aspect of modern culture; games are used for entertainment, education, training and more, and their impact on society is very large. This is driving a need for structured principles and practices for developing games and simulations. Also, reducing the cost and difficulties of making games could greatly benefit some "fringe" game developers, such as independent game developers, serious game developers, and even research game developers, who traditionally have neither the budget nor the manpower to tackle some of the challenges associated with making a modern game.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Modern games often intersect with databases, both when constructing the core of the simulation and managing the massive amounts of (often distributed) information associated with a game. By building awareness of this relationship we hope to encourage more database researchers to help share the "wisdom of their trade" with game developers, creating a fruitful exchange of knowledge and offering new viewpoints to older problems.

References

1. Entertainment Software Association: Industry Facts. (2010)
2. Ritterfeld, U., Cody, M., Vorderer, P.: Serious Games: Mechanisms And Effects. (2009)
3. Conway, M., Pausch, Y., Gossweiler, R., Burnette, T.: Alice: A Rapid Prototyping System for Building Virtual Environments. (1995)
4. Buckland, M.: Programming Game AI by Example., Sudbury, MA (2004)
5. Giuseppe Maggiore, M.: Monadic Scripting in F# for Computer Games., Oslo, Norway (2011)
6. Maggiore, G., Spanò, A., Orsini, R., Costantini, G., Bugliesi, M., Abbadi, M.: Designing Casanova: a language for games. In Proceedings of the 13th conference on Advances in Computer Games, ACG 13, Tilburg, 2011, Springer. In : 13th Internation Conference Advances in Computer Games (ACG), Tilburg, Netherlands (2011)
7. Maggiore, G., Bugliesi, M., Orsini, R.: Casanova Papers. In: Casanova project page. (Accessed 2011) Available at: http://casanova.codeplex.com/wikipage?title=Papers
8. Walker White, A.: Scaling games to epic proportions. In : Proceedings of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD), New York, NY, USA, p.31–42 (2007)
9. Perez, S., Sarris, A.: Technical Report for IRDS Conceptual Schema, Part 1: Conceptual Schema for IRDS, Part 2: Modeling Language Analysis. (1995)
10. Garcia-molina, H., Ullman, J., Widom, J.: Database System Implementation. (1999)
11. Weikum, G., Vossen, G.: Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery. (2001)
12. Bernier, Y.: Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization. https://developer.valvesoftware.com/wiki/Latency_Compensating_Methods_in_Client/Server_In-game_Protocol_Design_and_Optimization (2001)
13. Trelford, P.: Learning with F#. In : Proceedings of the 4th ACM SIGPLAN workshop on Commercial users of functional programming (2007)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

On the Use of Dimension Properties in Heterogeneous Data Warehouse Integration*

Marius-Octavian Olaru and Maurizio Vincini

Department of Information Engineering - DII
University of Modena, Italy
{mariusoctavian.olaru, maurizio.vincini}@unimore.it

Abstract. A new trend in Business Intelligence is the process of combining information from two or more different and heterogeneous Data Warehouses. Existing solutions rely mostly on the Extract-Transform-Load (ETL) approach, a costly and laborious process. The process of Data Warehouse integration can be greatly simplified by developing methods to semi-automatically discover semantic mappings among attributes of two or more different, heterogeneous Data Warehouse schemas, like the one proposed in this paper.

1 Introduction

The dynamic economical context that characterizes today's economical markets has increased the number of cases where companies need to integrate information coming from two or more heterogeneous, independently developed, Data Warehouses. For example, it is common for two companies to merge, or for one company to acquire one or more other companies; in both cases, the independent DWs must be integrated in order to offer management a unified view over the entire available information. A manual process of DW integration, base on ETL procedures, is laborious, time consuming and itself prone to errors. An automatic or semi-automatic method can increase the efficiency of such process. This has been proved in the data integration area where designers make use of semi-automatic tools (like [2]) as support for the mapping discovery process between independent and heterogeneous data sources. The problem of heterogeneous DW integration can be seen as a subcase of data integration, as the information to be integrated is multidimensional, that is why the authors believe that a specific solution that takes into account this particularity may yield better results than classical data integration techniques.

In this paper, we propose a semi-automatic method to discover mappings among Data Warehouses dimensions that exploits topological properties of the DW dimension levels together with semantic annotation techniques. In particular, we rely on graph theory and the normalization *Combined Wordsense Disambiguation* (CWSD) techniques developed inside the MOMIS data integration project[14].

This paper is organized as follows: Section 2 presents an overview on related work, Section 3 provides a description of the proposed method, meanwhile Section 4 draws the conclusions and presents the future work.

* This work has been published in [7].

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2 Related Work

Until now there have been few attempts to formalize the problem of Data Warehouse Integration, and no complete solution has been yet proposed. Formalization approaches include those proposed by Kimball in [10] or the work presented in [8, 15], where the authors define the concept of *conformed dimensions*, from a theoretical point of view.

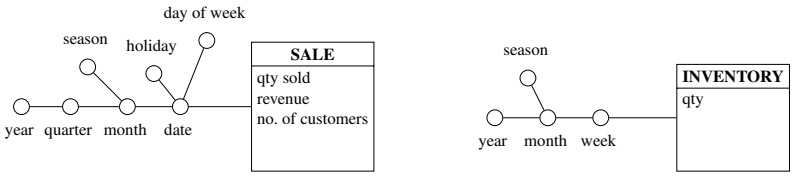
A solution to the problem is proposed in [9], where an architecture for the exchange of multidimensional information (called BIN-*Business Intelligence Network*, or *Peer-to-Peer Data Warehouse*) is proposed, and the OLAP query reformulation problem in the proposed architecture is formalized. A Peer-to-Peer Data Warehouse is a network in which the local peer has the capability of executing queries over the local Data Warehouse and to forward them to the network. The queries are then rewritten against the remote compatible multidimensional models using a set of mapping predicates between the local schema and the remote schema. The main problem of this architecture is that the mapping predicates have to be manually generated, so the approach is not scalable.

In [1] there is an attempt to automate the mapping process between multidimensional structures. The authors present a method, based on early work in data integration [4, 6], that allow designers to automatically discover schema mappings between two heterogeneous Data Warehouses. The *class similarity* (or *affinity* concept, as described in [5]) is used to find similar attributes (*facts*, *dimensions*, *aggregation levels* and *dimensional attributes*) and similarity functions for multidimensional structures based on that concept are proposed. Our method makes use of semantics, but with a different approach: [1] relies on semantic relations to discover the mappings among attributes, whereas we exploit them only as a validation technique.

3 Mapping Discovery

We propose an instance-level[12] technique for the semi-automatic discovery of mappings between heterogeneous DW dimension levels that exploits semantics and the specific graph-like structure that the partial order relationship imposes on the dimensional attributes.

Although may contain different information, dimensional hierarchies representing the same concept usually maintain the same structure. Consider, for example, that the time dimension in the first schema of Figure 1 (S_1) contains all the *days* between January



(a) SALE(S_1) (b) INVENTORY(S_2)

Fig. 1. Sample schemas

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

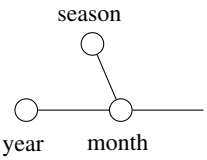


Fig. 2. A common *sub-dimension*

1^{st} 2008 and December 31^{st} 2010 (three complete years), meanwhile the time dimension from the second schema S_2 contains all the *weeks* between January 1^{st} 2010 and December 31^{st} 2011 (two complete years). A simple data intersection will reveal that the attributes values sets are only partially overlapped, due to the fact that the dimensions cover different time periods, and to the fact that the dimensions contain distinct attributes (for example, S_1 contains the dimensional level *quarter*, meanwhile S_2 contains the dimensional level *week*). However, some common topological properties are preserved among the two dimensions. For example, a *year* is an aggregation of 12 different *months* in both schemas; similarly, a *season* is composed of 3 different *months*. These common properties may be used to identify similar elements in the two dimension levels, and to generate mappings that are able to express the exact relationships among them.

The dimension in Figure 2 can thus be seen as a common *sub-dimension*[1] of the two initial dimensions that can be used to semi-automatically map levels of S_1 to levels of S_2 and *vice-versa*.

The method propose in this paper can be summarized in this steps:

1. *Candidate mappings set generation*: pairs of equivalent nodes, together with a set of 5 rules are used to generate the list of candidate mappings.
2. *Semantic validation*: the candidate mappings are then validated using a semantic approach.

3.1 Candidate Mappings Generation

As mappings predicates, we used a subset of those proposed in [9]. In particular, we considered the *equi-level, roll-up, drill-down* and *related* mapping predicates.

For the computation of the common sud-dimension, we simply consider the dimensions as directed labeled graphs, where the label of each edge is the *cardinality ratio*[2] among two different dimension levels. Using graph theory, it is possible to compute a maximum-rank common subgraph, and to identify pairs of nodes of the two initial graphs that correspond to the same node in the subgraph, that are called *equivalent*. In the example in Figure 1, the nodes $S_1.month$ and $S_2.month$ are equivalent, as are the nodes $S_1.year$ and $S_2.year$, and $S_1.season$ and $S_2.season$.

[1] A *sub-dimension* is intended as a new dimension obtained by removing one or more aggregation levels

[2] By *cardinality ratio* we simply intend the ratio among the number of different elements contained in different aggregation levels.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

We discover mappings by exploiting the following 5 rules:

Let P_x and P_y be two nodes of the first dimension such that there is a path from P_x to P_y, and P_h and P_k two nodes of the second dimension such that there is a path from P_h and P_k

1. **Rule 1**: If P_x and P_h are equivalent, add the mappings:
   * $P_x\ (equi-level)\ P_h$
2. **Rule 2**: if $P_x\ (equi-level)\ P_h$, add the mappings:
   * $P_y\ (roll-up)\ P_h$
   * $P_h\ (drill-down)\ P_y$
3. **Rule 3**: if $P_y\ (equi-level)\ P_h$, add the mappings:
   * $P_x\ (drill-down)\ P_h$
   * $P_h\ (roll-down)\ P_x$
4. **Rule 4**: if $P_y\ (equi-level)\ P_h$, add the mappings:
   * $P_x\ (drill-down)\ P_k$
   * $P_k\ (roll-up)\ P_x$
5. **Rule 5**: for every nodes P_x and P_h of the two graphs for which there has not been found any mapping rule, add the mapping:
   * $P_x\ (related)\ P_h$

Using these simple rules, it is possible to generate a complete set of mapping candidates that express the relationship among every two elements of the two schemas. For example, Rule 1 will produce a mapping $S_1.month\ (equi-level)\ S_2.month$, while Rule 3 will produce the rule $S_1.month\ (roll-up)\ S_2.week$[3].

3.2 Semantic Mapping Validation

To validate the discovered mappings, we decided to weight and prune them using a semantic approach based on Lexical Annotation. Lexical Annotation is the process of explicit assignment of one or more meanings to a term w.r.t. a thesaurus. To perform lexical annotation, we exploited the CWSD[3, 13] (*Combined Word Sense Disambiguation*) algorithm implemented in the MOMIS Data Integration System which associates to each element *label* (i.e., the name of the element) one or more meanings w.r.t. the WordNet lexical thesaurus [11].

One important issue in Data Warehousing is that often the schema and attribute labels are abbreviated, or they are Compound Nouns (CN). This makes it difficult for an automatic algorithm to assign a semantic meaning w.r.t. a thesaurus, like WordNet. The main reason for which we chose the algorithm in [14] is that it is able to increase the accuracy of the annotation process by doing both context-aware abbreviation expansion and CN disambiguation.

Starting from lexical annotations, we can discover semantic relations among attributes of the different Data Warehouses by navigating the wide semantic network of WordNet. In particular, the WordNet network includes[4]: *synonim*, *hypernim* and

[3] For the sake of simplicity, we do not present the entire mapping set.

[4] WordNet includes other semantic and lexical relations such as antonym, cause etc. which are not relevant for our approach

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Table 1. Coefficient Assignment

P_i/P_j	equi-level	roll-up	drill-down	related
same/synonyms	1	0.7	0.7	0.7
hypernyms	0.9	0.7	1	0.8
hyponyms	0.9	1	0.7	0.8
correlated terms	0.7	0.7	0.7	1
holonyms	0.7	1	0.3	0.8
meronyms	0.7	0.3	1	0.8

meronym relationships. We added the *correlated terms* relation that can be directly derived by WordNet: two terms are correlated if they are connected by a hyponym or meronym relation to the same WordNet synset. Thus, for each identified mappings, we first annotate each label by using the algorithm in [14] and then, we discover the shortest path of semantic relations connecting the two elements into the WordNet network. The goal is to validate the mappings by computing for each of them a similarity weight on the basis of the identified WordNet paths. We computed the similarity weight by assigning to every edge (i.e., WordNet relation) of the path a coefficient using the assignment rules in Table 1. The final similarity weight is given by the product of the single coefficients (thus, long paths will usually have lower similarity weights than short or direct paths). These coefficients were defined by considering that an *equi-level* is semantically similar to a *same/synonim* relationship in WordNet, *roll-up* is similar to a *hyponim* or to a *holonym*, meanwhile a *drill-down* relationship is similar to the *hypernym* or *meronym* relationship. For the other mapping/relationship combinations we associated coefficients (lower than 1) on the basis of their relevance. For example, to the combination *drill-down/holonym*, we associated a low coefficient (0.3) as they semantically represent opposite concepts. Starting from these computed coefficients, we can prune the discovered mappings, by applying a coefficient threshold.

4 Conclusions and Future Work

In this paper, we argued that topological properties of dimensions in a Data Warehouse can be exploited to find semantic mappings between two or more different Data Warehouses. We showed how these properties can be used in conjunction with semantic techniques to efficiently generate a mapping set between the attributes of the dimensions of two independent Data Warehouses.

Our method is effective for the integration of Data Warehouses in the case of reasonable complete DW instances. If partial/scarce information is present in the DWs, then the cardinality ratio among levels might vary rendering the mapping generation step inefficient.

Another important observation we made during our research is that the mapping predicates have no exact correspondence with the WordNet semantic relations, so it is impossible to assign an exact semantic weight coefficient to a specific type of mapping. These semantic weights depend on the context of the Data Warehouse. In the future,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

we plan to investigate on how the fine tuning of these coefficients can affect the accuracy of the mapping method. In any case, for maximum accuracy a human validation is required. This is also an issue in data integration where developers/analysts rely on semi-automatic tools to discover semantic correspondences, but unfortunately the process cannot be entirely automatic if maximum accuracy is required. As any approach proposed so far in Data Warehouse integration has flaws, we believe that a combination of approaches (like the topological/semantic approach proposed in this paper) could improve the accuracy of the mapping discovery process.

References

1. Banek, M., Vrdoljak, B., Tjoa, A.M., Skocir, Z.: Automated Integration of Heterogeneous Data Warehouse Schemas. IJDWM **4**(4) (2008) 1–21
2. Beneventano, D., Bergamaschi, S., Gelati, G., Guerra, F., Vincini, M.: MIKS : An Agent Framework Supporting Information Access and Integration. In Klusch, M., Bergamaschi, S., Edwards, P., Petta, P., eds.: AgentLink. Volume 2586 of Lecture Notes in Computer Science., Springer (2003) 22–49
3. Bergamaschi, S., Bouquet, P., Giacomuzzi, D., Guerra, F., Po, L., Vincini, M.: An Incremental Method for the Lexical Annotation of Domain Ontologies. Int. J. Semantic Web Inf. Syst. **3**(3) (2007) 57–80
4. Bergamaschi, S., Castano, S., Vincini, M.: Semantic Integration of Semistructured and Structured Data Sources. SIGMOD Record **28**(1) (March 1999) 54–59
5. Bergamaschi, S., Castano, S., Vincini, M., Beneventano, D.: Retrieving and integrating data from multiple sources: the MOMIS approach. Data Knowl. Eng. **36**(3) (2001) 215–249
6. Bergamaschi, S., Guerra, F., Vincini, M.: A Peer-to-Peer Information System for the Semantic Web. In: AP2PC. (2003) 113–122
7. Bergamaschi, S., Olaru, M.O., Sorrentino, S., Vincini, M.: Semi-automatic Discovery of Mappings Between Heterogeneous Data Warehouse Dimensions. In: International Conference on Advances in Communication and Information Technology, Amsterdam, ACEEE (2011)
8. Cabibbo, L., Torlone, R.: On the Integration of Autonomous Data Marts. In: SSDBM, IEEE Computer Society (2004) 223–
9. Golfarelli, M., Mandreoli, F., Penzo, W., Rizzi, S., Turricchia, E.: Towards OLAP query reformulation in Peer-to-Peer Data Warehousing. In Song, I.Y., Ordonez, C., eds.: DOLAP, ACM (2010) 37–44
10. Kimball, R., Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. Volume 32. John Wiley & Sons, Inc., New York, NY, USA (2002)
11. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: WordNet: An on-line lexical database. International Journal of Lexicography **3** (1990) 235–244
12. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal **10**(4) (2001) 334–350
13. Sorrentino, S., Bergamaschi, S., Gawinecki, M.: NORMS: An automatic tool to perform schema label normalization. In: ICDE. (2011) 1344–1347
14. Sorrentino, S., Bergamaschi, S., Gawinecki, M., Po, L.: Schema label normalization for improving schema matching. Data & Knowledge Engineering **69**(12) (December 2010) 1254–1273
15. Torlone, R.: Two approaches to the integration of heterogeneous data warehouses. Distributed and Parallel Databases **23**(1) (2008) 69–97

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Structural and Content Queries on the Nested Sets Model

Gianmaria Silvello

Department of Information Engineering, University of Padua, Italy
silvello@dei.unipd.it

Abstract. Hierarchical structures are pervasive in computer science. They are a fundamental means for modeling many aspects of reality and for managing wide corpora of data and digital resources. How to model and manage data hierarchies is a major theme in database theory and practice. The most important hierarchical structure is the tree, which has been widely studied, analyzed, and adopted in several contexts and scientific fields over time. In this paper we describe an alternative approach for modeling and managing hierarchies, called the Nested Sets Model (NS-M) and we describe how it is possible to reconsider structural and content queries by exploiting this model.

1 Motivation

Hierarchical structures are pervasive in computer science. They are a fundamental means for modeling many aspects of reality and for managing wide corpora of data and digital resources. How to model and manage data hierarchies is a major theme in database theory as well as in database applications [1] and it has been extensively treated in the context of database abstractions, the relational model, the complex value model, object-oriented database systems, and semistructured data and *Extensible Markup Language* (XML).

In the database field and more generally in computer science, the fundamental structure used to model and represent hierarchies is the tree. The tree has been formalized and deeply studied, its properties are well understood, and many of the algorithms based on it run in polynomial time. Therefore, the tree has been exploited by researchers and developers in many different fields to model and solve their problems. In this sense, trees are considered "*the most important nonlinear structures that arise in computer science*" [6].

In this article we propose an alternative approach for modeling and managing hierarchies, called Nested Sets Model (NS-M), which is based on the inclusion between sets as a means to capture hierarchical relations. The foundational idea underlying this model is to express the hierarchical relationships between objects through the inclusion property between sets, in place of the binary relation between nodes exploited by the tree. Therefore, the pivotal question we want to address is how structural and content queries on hierarchies can be handled with a data model other than the tree. In order to answer this question we define the

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

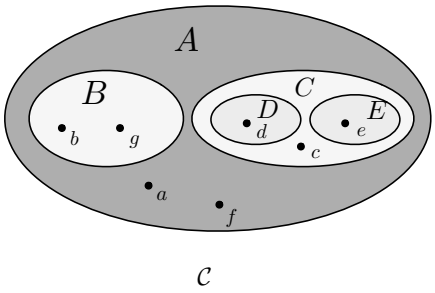


Fig. 1. A sample Nested Sets Collection (NS-C) mapped from a tree represented by means of an Euler-Venn diagram.

NS-M (Section 2.1), we describe how it allows us to handle structural and content components of a hierarchy (Section 2.2) and we prove its properties (Section 2.3). In Section 3 we specify basic structural and content query operations by highlighting how they can be handled by the NS-M; and, in Section 4 we draw some final remarks and future works.

2 The Nested Sets Model (NS-M)

2.1 Definition of the Model

The NS-M is formally defined as a collection of subsets where specific conditions must hold [2,4].

Definition 1 *Let A be a set and let $\mathcal{C}$ be a collection of subsets of A. Then $\mathcal{C}$ is a **Nested Sets Collection** (NS-C) if:*

$$A \in \mathcal{C}, \tag{2.1}$$

$$\forall H, K \in \mathcal{C} \mid H \cap K \neq \emptyset \Rightarrow H \subseteq K \vee K \subseteq H. \tag{2.2}$$

Therefore, we define a NS-C as a collection of subsets where two conditions must hold. The first condition (2.1) states that set A which contains all the subsets of the collection must belong to the NS-C itself. The second condition states the intersection of every couple of sets in the NS-C is not the empty-set only if one set is a proper subset of the other one. [1]

The NS-C is represented by means of an Euler-Venn diagram as we can see in Figure 1, which represents a sample NS-C composed of five nested sets:

[1] The graphical representation of a tree as a collection of nested sets was originally presented in [6] with no formal definition of a "nested sets model"; afterwards, this representation has been exploited in [3] to explain an alternative way to solve recursive queries over trees by using an integer intervals encoding. Also [5] proposed the same idea to solve recursion in relational databases.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

$\mathcal{C} = \{A, B, C, D, E\}$. We can see that A is the *top set* of $\mathcal{C}$ (i.e. the common superset of all the sets in $\mathcal{C}$) and thus Condition 2.1 is respected, and all the sets are either disjoints or one is a proper subset of the other as required by Condition 2.2.

2.2 Separating Structure and Content

From the *structural point-of-view*, a collection of subsets is represented by the sets in the collection and their inclusion dependencies. A collection of subsets at the *intensional level* is defined by its structure. Let us consider an example, we can say that $\mathcal{C} = \{A, B, C\}$ where $B \subseteq A$, $C \subseteq A$ and $B \nsubseteq C \wedge C \nsubseteq B$ is a NS-C because it respects conditions 2.1 and 2.2 of Definition 1. In this way, we know the structure of the collection and we know which relationships hold between the sets.

When we consider a collection of subsets $\mathcal{C}$ from the the *content point-of-view*, it means that we refer to its *extensional level*. From the *content point-of-view* a collection of subsets $\mathcal{C}$ is represented by the *extension* of the sets composing it; the properties of the sets are then verified by inspecting the sets and verify the elements that they contain. In this case, we say that the content of a collection of subsets defines the *extension* of such a collection. Therefore, we can say that $\mathcal{C} = \{A, B, C\}$ where $A = \{a, b, c, d\}$, $B = \{b\}$ and $C = \{c, d\}$ is the extension of a NS-C. In the next example we can see a NS-C defined at the intensional level which at the extensional level is satisfied by two different NS-C.

Example 1 *Let us consider the following NS-C defined at the intensional level:* $\mathcal{C} = \{A, B, C, D\}$ *where* $B \subseteq A$, $C \subseteq A$, $D \subseteq C$ *and* $B \nsubseteq C \wedge C \nsubseteq B$. *Then,* $A = \{a, b, c, d, e\}$, $B = \{b\}$, $C = \{c, d, e\}$, $D = \{d, e\}$ *is a valid instance for* $\mathcal{C}$, *as well as* $A = \{a, b, c, d, e, f\}$, $B = \{c, d\}$, $C = \{b, e, f\}$ *and* $D = \{f\}$; *indeed, they both satisfy the specified structural conditions.*

Both the structural and the content point-of-view are important for the treatment of the NS-M. We exploit the structure defined at the intensional level to define the properties of NS-M; whereas we exploit the extensional level to perform set operations which manipulate the content of the subsets composing the collections.

In the following we make extensive use of the concepts of collection of proper subsets and supersets and of direct subsets and supersets. Let $\mathcal{C}$ be a collection of sets and $A \in \mathcal{C}$ be a set, we define:

- $\mathcal{S}^-(A) = \{B \in \mathcal{C} \ : \ A \subset B\}$ to be the **collection of proper supersets** of A in $\mathcal{C}$;
- $\mathcal{S}^+(A) = \{B \in \mathcal{C} \ : \ B \subset A\}$ to be the **collection of proper subsets** of A in $\mathcal{C}$.
- $\mathcal{D}^-(A) = \{B \in \mathcal{C} \ : \ \big((A \subset B) \wedge (\nexists E \in \mathcal{C} \mid A \subset E \subset B)\big)\}$ to be the **collection of direct supersets** of A in $\mathcal{C}$.
- $\mathcal{D}^+(A) = \{B \in \mathcal{C} \ : \ \big((B \subset A) \wedge (\nexists E \in \mathcal{C} \mid B \subset E \subset A)\big)\}$ to be the **collection of direct subsets** of A in $\mathcal{C}$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2.3 Properties of the Model

Many properties of the NS-M are derived by the straightforward application of set theory as we show in the following example which takes into account the intensional level.

Example 2 *Let $\mathcal{C}$ be a NS-C. For all $H, K \in \mathcal{C} \mid H \subseteq K$ we can derive from set theory that $H \cup K = K$ and $H \cap K = H$. As well as we can say that for all $H, K \in \mathcal{C} \mid H \nsubseteq K \wedge K \nsubseteq H \Rightarrow H \setminus K = H \wedge K \setminus H = K$.*

In this example we see that the sets in a NS-C behave exactly as one would expect under the operations of union, intersection and set difference. Let us see an example which shows how these operations behave at the extensional level.

Example 3 *Let $\mathcal{C} = \{A, B, C\}$ be a NS-C, where $B \subseteq A$ and $C \subseteq B$. Then let us consider the following instance: $A = \{a, b, c, d, e\}$, $B = \{c, d, e\}$ and $C = \{e\}$. Then, $B \cup C = \{c, d, e\} = B$ and $B \cap C = \{e\} = C$.*

Let us consider a NS-C $\mathcal{C}$, the next proposition shows that for all $H \in \mathcal{C}$, H has at most one direct superset.

Proposition 1 *Let $\mathcal{C}$ be a NS-C. Then, $\forall H \in \mathcal{C}, |\mathcal{D}^-(H)| \leq 1$.*

Proof. Ab absurdo suppose that $\exists H \in \mathcal{C}$ such that $|\mathcal{D}^-(H)| > 1 \Rightarrow \exists K, L \in \mathcal{D}^-(H) \mid H \subseteq K \wedge H \subseteq L \wedge L \nsubseteq K \wedge K \nsubseteq L \Rightarrow K \cap L = H \Rightarrow \mathcal{C}$ is not a NS-C (condition 2.2 of Definition 1).

The following corollary to this proposition shows that the set with minimum cardinality in the collection of supersets of H is its direct superset.

Corollary 2 *Let $\mathcal{C}$ be a NS-C, $H \in \mathcal{C}$ be a set, $\mathcal{S}^-(H)$ be the collection of proper supersets of H and $K \in \mathcal{S}^-(H)$ where $\forall L \in \mathcal{S}^-(H), |K| \leq |L|$ be the subset with minimum cardinality in $\mathcal{S}^-(H)$. Then, $\mathcal{D}^-(H) = K$.*

Proof. We know from Proposition 1 that $|\mathcal{D}^-(H)| \leq 1$. Then, ab absurdo suppose that $\forall L \in \mathcal{S}^-(H), |K| \leq |L|$ and that $\exists W \in \mathcal{S}^-(H) \mid \big((|W| > |K|) \wedge (\mathcal{D}^-(H) = W)\big)$. This means that $H \subseteq W \wedge H \subseteq K$ and by definition of NS-M $W \subseteq K \vee K \subseteq W$. If $W \subseteq K \Rightarrow |W| < |K|$; if $K \subseteq W \Rightarrow |K| < |W|$. So if $\mathcal{D}^+(H) = W \Rightarrow |W| < |K|$.

The next proposition proves that the direct subsets of H are always disjoints.

Proposition 3 *Let $\mathcal{C}$ be a NS-C and $H \in \mathcal{C}$ be a set, then $\forall K, L \in \mathcal{D}^+(H), K \cap L = \emptyset$.*

Proof. Ab absurdo suppose that $K \cap L \neq \emptyset \Rightarrow K \cap L = W$ such that $|W| \geq 1 \wedge W \nsubseteq K \wedge W \nsubseteq L \Rightarrow \mathcal{C}$ is not a NS-C.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

3 An Insight on Structural and Content Queries

When we deal with a NS-C, we can distinguish between **structural** and **content** queries. Structural queries ask for the relationships between the sets in a collection of subsets – e.g. "Return all the sets in $\mathcal{C}$ which are supersets of the set $H \in \mathcal{C}$", in this case we are not interested in the actual content of the sets, instead we just want to know which sets are supersets of H. Content queries ask for the actual content (the elements) of the sets in a collection of subsets – e.g. "Return all the elements in $\mathcal{C}$ belonging to supersets of $H \in \mathcal{C}$" or "Return all the elements in $\mathcal{C}$ belonging to subsets of $H \in \mathcal{C}$". A possible data structure to implement these queries has to take into account the structural and the content components of the NS-M. it is possible to propose a dictionary-based data structure which from the structural point-of-view, stores the inclusion dependencies between the sets; and, from the content point-of-view, it stores the materialization of the sets (i.e. the elements of each set).

If we consider a NS-C $\mathcal{C}$ and a set $H \in \mathcal{C}$, the **structural** query operations we point out are: Descendants(H) which returns the sets which are descendants of H, Ancestors(H) which returns the sets which are ancestors of H, Children(H) which returns the sets which are children of H, and Parent(H) which returns the set which is the parent of H. The worst-case scenario for all these structural queries is represented by a NS-C structured as a chain. In the case of the Descendants(H) query the worst-case input set H is the top set because the query has to return all the sets in the given NS-C. All other structural query operations can be implemented to run in constant time for whichever collection of subsets and input set by exploiting the described properties of NS-M. For instance, the Parent operation can be implemented in $O(1)$ time by exploiting the fact that every set in a NS-C has at most one direct superset as proved in Proposition 1.

The **content** query operations are: Elements(H) which returns the elements in the set H, DescendantElements(H) which returns the elements in the descendants of H, AncestorElements(H) which returns the elements in the ancestors of H, ChildrenElements(H) which returns the elements in the children of H, and ParentElements(H) which returns the elements in the parent of H.

The NS-M is built in such a way that each set contains all the elements of its descendants; therefore, it is possible to answer this query without browsing the collection of subsets (without browsing the hierarchy) or without passing through any structural query. We can see how these queries are independent with respect to their correspondent structural queries. In order to answer the content queries we do not need to browse the hierarchical structure of the collection of subsets.

4 Final Remarks and Future Works

By defining the NS-M we showed that it is possible to model hierarchies exploiting the inclusion property between sets in place of the binary relationship

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

between nodes adopted by the tree. We provided the theoretical basis for employing the NS-M as an alternative to the tree to model hierarchies, and for exploiting a set-theoretical environment for the definition of problems and solutions with hierarchies. It is possible to exploit the fact that content and structure components of hierarchies are modeled as independent building blocks of the NS-M to define structural and content queries. We gave a first insight from the theoretical point-of-view on how content queries do not strongly rely on structural queries as happens with the tree.

In the future we are going to conduct experiments on synthetic and real datasets to experimentally verify that both structural and content query operations are scalable and efficient in the NS-M. The experimental analysis will be aimed to show how the choice between the tree and the NS-M has to be made on an application basis by evaluating which operations are executed more frequently. It will be thus possible to investigate and propose data structures optimized for specific contexts and applications so that to gain further performances for the NS-M.

Acknowledgments

The author thanks Maristella Agosti and Nicola Ferro which supported this work and contributed to many of the presented ideas. The author would like to thank Carlo Meghini for his useful and accurate observations regarding the formalization and the properties of the NS-M. The PROMISE network of excellence[2] (contract n. 258191) projects, as part of the 7th Framework Program of the European Commission, have partially supported the reported work.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. M. Agosti, N. Ferro, and G. Silvello. The NESTOR Framework: Manage, Access and Exchange Hierarchical Data Structures. In *Proceedings of the 18th Italian Symposium on Advanced Database Systems*, pages 242–253. Società Editrice Esculapio, Bologna, Italy, 2010.
3. J. Celko. *Joe Celko's SQL for Smarties: Advanced SQL Programming*. Morgan Kaufmann, San Francisco, California, USA, 2000.
4. N. Ferro and G. Silvello. The NESTOR Framework: How to Handle Hierarchical Data Structures. In M. Agosti, J. Jose Borbinha, S. Kapidakis, C. Papatheodorou, and G. Tsakonas, editors, *Proc. 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2009)*, pages 215–226. Springer, Heidelberg, Germany, 2009.
5. M. Kamfonas. Recursive Hierarchies: The Relational Taboo! *The Relational Journal*, October/November 1992.
6. D. E. Knuth. *The Art of Computer Programming, third edition*, volume 1. Addison Wesley, Reading, MA, USA, 1997.

[2] `http://www.promise-noe.eu/`

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Knowledge-based Real-Time Car Monitoring and Driving Assistance

Michele Ruta, Floriano Scioscia, Filippo Gramegna,
Giuseppe Loseto, and Eugenio Di Sciascio

Politecnico di Bari, via Re David 200 I-70125 Bari, Italy,
m.ruta@poliba.it, f.scioscia@poliba.it, gramegna@deemail.poliba.it,
loseto@deemail.poliba.it, disciascio@poliba.it

Abstract. This paper describes a knowledge-based framework for a driving assistance via smartphone. Vehicle information extracted through On Board Diagnostics (OBD-II) protocol, data acquired from smartphone embedded micro-devices and information retrieved from the Web are properly combined. Data fusion and classification algorithms allow to identify and annotate relevant contexts and events in real time and semantic-based matchmaking is exploited to infer dysfunctional situations.The proposed approach has been implemented in an Apple iPhone application, and evaluated in real-world test drives.

Keywords: Semantic Web, Ubiquitous Computing, Data Fusion, On-Board Diagnostics, Intelligent Transportation Systems

1 Introduction

Modern vehicles are equipped with several Electronic Control Units (ECUs) coordinating and monitoring internal components and subsystems, communicating over one or more car network buses. In particular, international regulations today mandate all new vehicles must support the *On Board Diagnostics, version 2* (OBD-II) protocol (http://www.arb.ca.gov/msprog/obdprog/obdprog.htm) and be equipped with an OBD-compliant interface to provide direct and standard access to data in the internal automotive network. Furthermore, in case of malfunctions, Diagnostic Trouble Code (DTC) values are stored in the car ECU and can be later retrieved by maintenance technicians using proper tools. Recently, access has been granted also to the general public of car enthusiasts by the development of OBD-II *scan tools*, cheap electronic devices that bridge the OBD-II port with standard wired (RS-232, USB) or wireless (Bluetooth, IEEE 802.11) computer communication interfaces.

This paper enhances framework in [1], able to interpret vehicle data extracted via OBD-II, integrate environmental information and detect potential risk factors. Beside providing warnings for that, now the system gives suggestions during driving and evaluates car efficiency and environmental impact. It is fully compliant with widely available smartphones.By means of properly devised processing and fusion algorithms, the system is able to identify and classify given

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

high-level events and conditions, based on low-level data streams. Furthermore, leveraging Semantic Web languages and technologies, events are annotated w.r.t. an ontology that models characteristics influencing driving safety and undergo a matchmaking process –exploiting an embedded matchmaker supporting non-standard reasoning services [2]. The matchmaking outcome is used to suggest the driver (via her smartphone) actions and behaviors she should adopt. The proposed framework has been implemented in a prototypical mobile software system, using the Apple iPhone[1] smartphone as reference platform. The experimental evaluation has been carried out in several real-world test drives under different conditions, evidencing both feasibility and usefulness of the approach.

In the remaining of the paper, after a survey on relevant related work in Section 2, the proposed framework is described in Section 3. Tests corroborating the approach are presented in Section 4. Conclusion and future work close the paper.

2 Related Work

Literature about OBD-based systems for vehicle monitoring and alert refers to *remote* and *on-board* solutions, respectively. In the former type [3, 4], GPS data and vehicle OBD DTCs are sent to a Maintenance Center server via GPRS/UMTS and stored into a database, which is scanned by a diagnostics expert system that generates a rough suggestion to advise the maintenance technicians. The proposal presented here does not require experts to evaluate system outputs. Furthermore, information processing refers to a smartphone application and then it better resembles an on-board approach. Consider that, though useful for managing vehicle fleets, remote monitoring does not allow a direct driver assistance. Several on-board monitoring prototypes and reporting systems have been already proposed [5, 6]. Nowadays freeware and commercial software packages are available, that allow to monitor OBD-II vehicle data by using just a smartphone and off-the-shelf scan tools. Nevertheless, to the best of our knowledge, all existing on-board monitoring systems directly display the acquired low-level data. They do not analyze the information to provide more meaningful and user-friendly indications, though researchers have widely acknowledged the possibility to exploit the wealth of real-time vehicle data available trough OBD in order to analyze driver behavior [7]. Current efforts aim to use multi-source information fusion to better interpret the relationships between driving habits and vehicle performance, as well as to detect risk situations [8]. Nevertheless, in available works analysis is performed off-line after data gathering, so they are not able to provide an automatic real-time driver support.

3 Framework

Starting from [1], the proposed system monitors a larger set of parameters, including environmental conditions, gas emissions, fuel consumption, engine load,

[1] iPhone Specifications, http://www.apple.com/iphone/specs.html

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

vehicle performance, safety equipments status and driving style. The *Kiwi Wifi*[2] wireless adapter interfaces car ECUs via OBD-II. When turned on, it builds an IEEE 802.11 ad-hoc network exposing a static IP address allowing an application to communicate with the OBD interface via socket in read/write mode. The proposed approach works along three subsequent stages: (i) data gathering; (ii) data fusion; (iii) semantic characterization and matchmaking. They are repeatedly executed; after each data gathering, further steps are executed and outcomes are displayed on the iPhone screen added to audible alerts.

Data gathering. As described in [1], OBD-II specifications only comprise the *Physical Signal Layer* (PSL) and *OBD-II Data Communication Layer* (DCL) of the ISO/OSI model. Though the system is able to retrieve all possible vehicle parameters via the OBD-II interface, our analysis focused on: coolant temperature, engine load, throttle position, RPM (Revolutions Per Minute), vehicle speed, MAF (Mass Air Flow), fuel level, upstream and downstream oxygen sensor. Considering wireless communication latency between the *Kiwi Wifi* PLX and the iPhone, the system requests a parameter every 0.3 seconds. Through the iPhone GPS receiver, the system gets latitude and longitude of vehicle current position, which are used to collect information about location address, road type and weather conditions via HTTP requests to Web services [1], using mobile Internet connectivity.

Data fusion. Previously collected data are processed at this stage to identify conditions and events to be annotated w.r.t. a reference ontology. The selected data fusion techniques provide adequate sensitivity for our purposes, while also maintaining moderate computational and memory requirements. Along with road features, driving style and traffic, as in [1], now the system can detect the following additional conditions. ***Emissions.*** To comply with emission regulations, OBD requires at least two oxygen sensors to evaluate the efficiency of catalytic converter for gasoline engines: an upstream and a downstream one. They provide feedback for maintaining the proper fuel mixture for efficient combustion and regular emissions, respectively. Each oxygen sensor produces a voltage value between 200mV and 900mV depending on the amount of hot oxygen it is exposed to. 15 samples at a time are considered also counting samples exceeding 900mV (MAX) and samples smaller than 500mV (MIN). This threshold was chosen because maximum voltage basically range from 500mV to 900mV, instead minimum voltages at most reach 500mV. In this way, for the upstream oxygen sensor, we infer that if $|MAX - MIN| < 5$, combustion is regular, otherwise not. For the downstream oxygen sensor, if $|MAX - MIN| > 9$, emissions are regular, otherwise not. ***Gear setting.*** A right use of gear lever is revealed measuring the ratio of throttle position (expressed in percent) to the engine load. Tests proved that in case of proper behavior the ratio value is around 1. A value greater than 2 indicates the user is requiring a power the engine cannot provide. The system computes the integral area of the throttle/engine load ratio considering 7 samples at a time. If the threshold value of 7.7 is exceeded, a wrong gear lever is detected. ***Fuel consumption and autonomy.*** Fuel consumption computation

[2] PLX Devices, Kiwi Wifi, http://www.plxkiwi.com/kiwiwifi/hardware.html

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

is based on MAF, *i.e.*, the mass of air (in grams) entering the engine every second. Then the volume of injected fuel (l/s) is:
$injected\_Fuel = MAF \div (fuel\_Factor \times fuel\_Density)[l/sec]$
where $fuel\_Factor$ is the stoichiometric ratio of air to fuel (equal to 14.7) and $fuel\_Density$ is the fuel density at 15°C (720 g/l for gasoline). Dividing vehicle speed by injected fuel, fuel consumption (in km/l) is obtained, from which average consumption and autonomy are computed.

Semantic annotation and matchmaking. The semantic annotation prepares the subsequent matchmaking phase. A toy ontology (not reported here for brevity) has been implemented in OWL-DL (Web Ontology Language, W3C Recommendation, http://www.w3.org/TR/owl-features/) formal language, grounded on Description Logics (DL) semantics. Their main classes are: **Vehicle** with subclasses *Safety_Equipment* and *Sensor*. **Safety_Equipment** is specialized in: *Fog_Lamp*, *ABS*, *ESP* and *Snow_Chains*. **Sensor** has subclasses: *Speedometer*, *RPM_Sensor*, *Fuel_Level_Sensor*, *MAF_Sensor*, *ThrottleLoad_Ratio*, *PreOxy_Sensor*, *PostOxy_Sensor*. Other classes are **Weather**, **Road_Surface**, **Road_Condition**, **Traffic**, **Driving_Style**. Main roles are: **hasDriving_Style**, **hasSafety_Equipment**.

While [1] relied on a remote matchmaking engine accessed via HTTP, here an Objective-C version of the Java mobile matchmaker described in [9] is exploited. This avoids dependence on cellular data connection and centralized reasoners, so improving significantly system reliability, latency and bandwidth consumption. The *Concept Abduction* non-standard inference was selected as reference reasoning service [2, 9]: in a nutshell, given a request R and a provided resource/service S, described w.r.t. a shared ontology, Concept Abduction allows to identify what is missing in S in order to completely satisfy R. In our framework, the semantic annotation of the context and the semantic description associated to an optimal vehicle functioning represent the request (*i.e.*, what factors are needed to travel safely and efficiently), while the semantic description of vehicle actual status and user driving style are the provided resource (*i.e.*, what factors are provided by the "vehicle+driver" system). Hence, the Abduction process will infer every requirement needed for a correct vehicle condition not explicitly satisfied by current vehicle configuration and driver behavior. Thus proper suggestions are provided to the driver in order to prevent dangers and/or reduce inefficiency. For example, let us suppose to monitor the level of vehicle exhaust gas emissions in a high density traffic situation. The semantic-based request for the system is (in DL formalism and w.r.t. the above cited ontology):

$$R \equiv Emission \sqcap \forall\, hasSensor\_PreOxy.Normal\_PreOxy \sqcap \forall\, hasSensor\_PostOxy.Normal\_PostOxy \sqcap Traffic \sqcap \forall\, hasSecure\_Device.ABS \sqcap \forall\, hasSensor\_Speed.Low\_Speed \sqcap \forall\, hasDriving\_Style.Even\_Pace\_Style.$$

It means that to have a low level emission and to avoid risks produced by an high density traffic, the output signals of upstream and downstream oxygen sensors should follow the above patterns, the vehicle should be equipped with ABS and the user should adopt an even pace driving style with low speed. Let us suppose the actual "vehicle+driver" system, *i.e.*, the provided resource is:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Fig. 1. Application user interface

$S \equiv Vehicle \sqcap \forall\, hasSensor\_PreOxy.Normal\_PreOxy \sqcap \forall\, hasSensor\_PostOxy.Abnormal\_PostOxy \sqcap \forall\, hasSecure\_Device.ESP \sqcap \forall\, hasSensor\_Speed.Low\_Speed \sqcap \forall\, hasDriving\_Style.Imprudent\_Style.$

By applying the abduction process to R and S:

$H \equiv \forall\, hasSensor\_PostOxy.Abnormal\_PostOxy \sqcap \forall\, hasSecure\_Device.ABS \sqcap \forall\, hasDriving\_Style.Even\_Pace\_Style.$

Hence, the system detects an abnormal output signal from the downstream oxygen sensor, indicating irregular exhaust gas emissions; moreover high traffic density suggests the need for ABS, as well as to adopt moderate drive. Since the vehicle only offers ESP and the user drives imprudently, the abduction outcome suggests to activate ABS and adopt an even pace driving style to decrease risks. Notice that, due to the Open World Assumption, semantic matchmaking –differently form database queries or rule-based engines– allows meaningful results to be inferred also in case of incomplete information (*e.g.*, lack of sensors on specific car models or temporary unavailability of data).

4 Experiments

The proposed framework was implemented in a system prototype for testing the approach.The test route was composed by: (i) a 3 km fast-flowing road; (ii) a 5 km fast-flowing stretch of an high speed road; (iii) a 2.5 km uphill stretch of a moderate speed road; (iv) a 3 km slow-flowing road. The test car, registered in 2001, is fueled with gasoline. At the application start-up, user can activate monitoring via the GUI whose main view is in Figure 1. After few seconds the icons in the *ADVICE* section are colored in function of matchmaking results. The application emits a single audio signal when an icon changes from green or yellow to orange, a double audio signal when it changes from orange to red so indicating different alert levels. Although quantitative performance measures were not taken at this stage of the work, system outcomes were globally coherent with the real driving situations and suggestions provided to the driver were appropriate. Warnings were sometimes issued with few seconds delay due the time needed to acquire the samples. These preliminary tests showed that

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

careful design and optimization allow semantic-based tools to be run in real-time on current mobile devices. By combining efficient data fusion algorithms and optimized semantic inference procedures, our approach can achieve both precision in condition detection and high-level information characterization for added-value driver support services.

5 Conclusion and Future Work

The paper presented a framework and a prototypical system for real-time vehicle monitoring and driving assistance. Information extracted via OBD-II, from smartphone micro-devices and Web services is used to annotate the context for further semantic-based inferences. Future work includes enhancements to the mobile prototype, *e.g.*, voice alerts and, as far as research is concerned, further OBD parameters and smartphone peripherals (*e.g.*, camera, microphone) could be used and concrete domains included in reference logic languages.

References

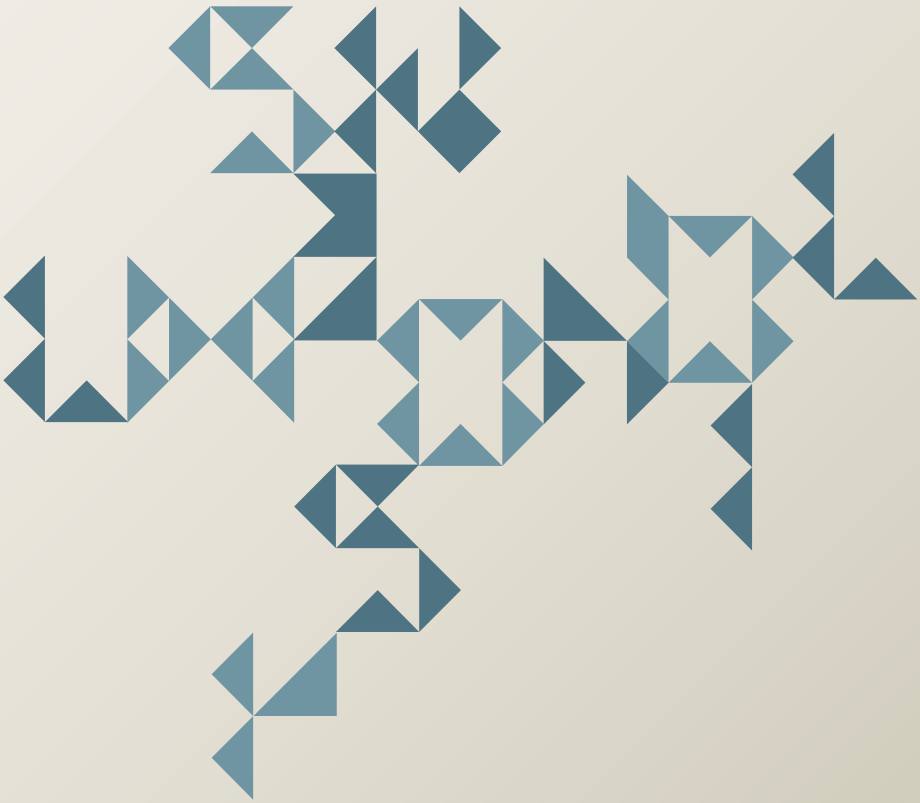
1. Ruta, M., Scioscia, F., Gramegna, F., Di Sciascio, E.: A Mobile Knowledge-Based System for On-Board Diagnostics and Car Driving Assistance. In: The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM2010), IARIA (2010) 91–96
2. Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., Tinelli, E.: A non-monotonic approach to semantic matchmaking and request refinement in e-marketplaces. International Journal of Electronic Commerce **12**(2) (2007) 127–154
3. Lin, C., Li, C.C., Yang, S.H., Lin, S.H., Lin, C.Y.: Development of On-Line Diagnostics and Real Time Early Warning System forVehicles. In: Sensors for Industry Conference, 2005. (2005) 45 –51
4. Lin, C.E., Shiao, Y.S., Li, C.C., Yang, S.H., Lin, S.H., Lin, C.Y.: Real-Time Remote Onboard Diagnostics Using Embedded GPRS Surveillance Technology. Vehicular Technology, IEEE Transactions on **56**(3) (2007) 1108–1118
5. Chen, Y., Xiang, Z., Jian, W., Jiang, W.: Design and implementation of multi-source vehicular information monitoring system in real time. In: Automation and Logistics, 2009. ICAL '09. IEEE International Conference on. (August 2009) 1771 –1775
6. Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., Dull, J., Sarkar, K., Klein, M., Vasa, M., Handy, D.: Vedas: A mobile and distributed data stream mining system for real-time vehicle monitoring. In Berry, M.W., Dayal, U., Kamath, C., Skillicorn, D.B., eds.: SDM, SIAM (2004)
7. Choi, S., Kim, J., Kwak, D., Angkititrakul, P., Hansen, J.: Analysis and Classification of Driver Behavior using In-Vehicle CAN-Bus Information. In: Biennial Workshop on DSP for In-Vehicle and Mobile Systems. (June 2007)
8. Quintero, M., Oñate Lopez, J., Rua, J., et al.: Intelligent erratic driving diagnosis based on artificial neural networks. In: ANDESCON 2010, IEEE (2010) 1–6
9. Ruta, M., Di Noia, T., Di Sciascio, E., Scioscia, F.: Abduction and Contraction for Semantic-based Mobile Dating in P2P Environments. In: Proceedings of 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, IEEE (2008) 626–632

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Author Index

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,



EDIZIONI LIBRERIA PROGETTO
ISBN: 978-88-96477-23-6