Actes de la conférence BDA 2017

Pierre Senellart, Amedeo Napoli

BDA 2017

Gestion de Données – Principes, Technologies et Applications

33e conférence
14-17 novembre 2017, Inria Nancy Grand-Est

Actes de la conférence BDA 2017
Conférence soutenue par Inria, le LORIA, le CNRS,
l'Université de Lorraine et le LIAS

Site de la conférence : `https://project.inria.fr/bda2017/`
Actes en ligne : `https://hal.inria.fr/BDA2017`

Message des organisateurs

La 33ème édition de la conférence sur la « Gestion de Données – Principes, Technologies et Applications » (BDA 2017) a eu lieu au centre Inria Nancy Grand-Est, à Nancy du 14 au 17 novembre 2017. Ces actes regroupent les versions courtes des articles présentés lors de cette conférence.

L'émergence du phénomène des données massives (Big Data) a pour effet de changer en profondeur la manière dont les différentes phases du processus d'acquisition et de valorisation des données sont mises en œuvre. Les données modernes sont volumineuses et souvent semi-structurées, incomplètes, imprécises, bruitées, inconsistantes, dynamiques ou fortement connectées. L'exploitation de ce type de données par des applications issues de domaines scientifiques et métier très variés pose de nombreux défis pour la communauté de recherche en informatique tant sur le plan fondamental qu'appliqué. Les technologies développées pour la gestion de données doivent également s'adapter à un environnement matériel et logiciel en perpétuelle mutation. Les infrastructures de stockage et de calcul ont évolué de manière importante ces dernières années. Les recherches autour du calcul haute performance (HPC) conduisent à la conception et à l'exploitation d'architectures nouvelles, massivement parallèles et distribuées. Les appareils mobiles, les architectures orientées services et la virtualisation sont omniprésents et sont en train de révolutionner la manière de fournir et d'utiliser les systèmes informatiques.

La recherche en gestion de données n'a jamais été aussi active, variée, ouverte sur d'autres champs de l'informatique et, au-delà, sur les grands défis des applications modernes.

Poursuivant la tradition de rencontres annuelles de la communauté de gestion de données francophone, BDA 2017 a invité académiques et industriels à soumettre leurs travaux récents pour rendre compte des défis et des avancées scientifiques et industrielles dans ce domaine en pleine effervescence.

Le programme scientifique a comporté 3 conférences invitées, 3 tutoriels, 22 présentations d'articles longs, 2 présentations d'articles courts, 5 démonstrations et 7 présentations de doctorants. Un événement co-localisé a également réuni les développeurs de la communauté openCypher. Une originalité de la conférence BDA est de proposer deux catégories pour les articles longs : articles originaux et articles déjà acceptés ou publiés dans une conférence internationale de renom. Cette deuxième catégorie a pour vocation d'attirer les meilleurs papiers de la communauté française et de donner à leurs auteurs l'opportunité de présenter leurs travaux devant la communauté de recherche nationale.

Nous tenons à remercier tous les auteurs pour leur excellente contribution, toute l'équipe d'organisation des journées BDA 2017, Pierre Bourhis et le comité de sélection des démonstrations et enfin tous les membres du comité de programme de BDA 2017.

Amedeo Napoli, CNRS, LORIA, Président du comité d'organisation
Pierre Senellart, ENS, Université PSL, Président du comité de programme

Table des matières

1 Comités de BDA 2017

Président des Journées
Amedeo Napoli (CNRS, LORIA)

Président du comité de Programme
Pierre Senellart (ENS, Université PSL)

Président du comité des démonstrations
Pierre Bourhis (CNRS, CRIStAL)

Comité de Programme
Reza Akbarinia (Inria Sophia)
Antoine Amarilli (Télécom ParisTech)
Laurent Amsaleg (CNRS, IRISA)
Marie-Aude Aufaure (CentraleSupélec)
Laure Berti-Équille (Qatar Computing Research Institute)
Luc Bouganim (Inria Saclay)
Stéphane Bressan (National University of Singapore)
Raja Chiky (ISEP)
Radu Ciucanu (Université Clermont Auvergne)
Emmanuel Coquery (Université Claude Bernard Lyon 1)
Jérôme Darmont (Université Lumière Lyon 2)
Bruno Defude (Télécom Sud Paris)
Stéphane Gançarski (Université Pierre-et-Marie-Curie)
Pierre Genevès (CNRS, LIG)
Ioana Ileana (Université Paris-Descartes)
Vincent Leroy (Université de Grenoble)
Silviu Maniu (Université Paris-Sud)
Zoltan Miklos (Université Rennes 1)
Joachim Niehren (Inria Lille)
Werner Nutt (Université libre de Bozen-Bolzano)
Nicoleta Preda (Université Versailles-Saint-Quentin)
Philippe Rigaux (CNAM)
Soror Sahri (Université Paris-Descartes)
Patricia Serrano-Alvarado (Université de Nantes)
Cristina Sirangelo (Université Paris-Diderot)
Olivier Teste (Université Paul Sabatier)
Michaël Thomazo (Inria Saclay)
Victor Vianu (Université California San Diego & Inria)
Agnès Voisard (FU Berlin and Fraunhofer FOKUS)

Comité des démonstrations
Sihem Amer-Yahia (CNRS, LIG)
Camille Bourgaux (TU Dresden)
Francesca Bugiotti (CentraleSupélec)
Sarah Cohen-Bouliaka (Université Paris-Sud)
David Gross-Amblard (Université Rennes 1)
Catherine Roussey (IRSTEA)
Federico Ulliana (Université de Montpellier)

2 Conférenciers invités

2.1 Ontology Mining by exploiting Machine Learning for Semantic Data Management
Claudia d'Amato

Bio : Claudia d'Amato obtained her PhD in 2007 from the University of Bari, defending the thesis titled "Similarity Based Learning Methods for the Semantic Web" for which she got the the nomination as author of one of the Best Italian PhD Thesis in Artificial Intelligence from the Artificial Intelligence Italian Commission for the AI*IA award 2007. She pioneered the research on developing Machine Learning methods for ontology mining, that still represents her main research interest. Her research activity has been disseminated through 19 journal papers, 12 book chapters, 55 papers in international collections, 27 papers in international workshop proceedings and 13 articles in national conference and workshop proceedings. She edited 27 books and proceedings and 3 journal special issues. During her research activity she also won several best paper awards. Claudia d'Amato served/is serving as Program Chair at ISWC 2017, ESWC 2014, Vice-Chair at ISWC'09, Journal Track chair at WWW 2018, Machine Learning Track Chair at ESWC'12-'13-'16-'17, PhD Symposium chair at ESWC'15 and Workshop and Tutorial Chair at ISWC'12, EKAW'12, ICSC'12. She served/is serving as a program committee member of a number of international conferences in the area of Artificial Intelligence, Machine Learning and Semantic Web such as AAAI, IJCAI, ECAI, ECML, ISWC, WWW, ESWC.

Abstract : In the Semantic Web view, ontologies play a key role. They act as shared vocabularies to be used for semantically annotating Web resources and they allow to perform deductive reasoning for making explicit knowledge that is implicitly contained within them. However, noisy/inconsistent ontological knowledge bases may occur, being the Web a shared and distributed environment, thus making deductive reasoning no more straightforwardly applicable. Machine learning techniques, and specifically inductive learning methods, could be fruitfully exploited in this case. Additionally, machine learning methods, jointly with standard reasoning procedure, could be usefully employed for discovering new knowledge from an ontological knowledge base, that is not logically derivable. The focus of the talk will be on various ontology mining problems and on how machine learning methods could be exploited for coping with them. For ontology mining are meant all those activities that allow to discover hidden knowledge from ontological knowledge bases, by possibly using only a sample of data. Specifically, by exploiting the volume of the information within an ontology, machine learning methods could be of great help for semi-automatically enriching and refining existing ontologies, for detecting concept drift and novelties within ontologies and for discovering hidden knowledge patterns. If on one hand this means to abandon sound and complete reasoning procedures for the advantage of uncertain conclusions, on the other hand this could allow to reason on large scale and to to dial with the intrinsic uncertainty characterizing the Web, that, for its nature, could have incomplete and/or contradictory information.

2.2 Swift Logics for Big Data
Georg Gottlob

Bio : Georg Gottlob is a Professor of Informatics at Oxford University and at TU Wien. His interests include KR, theory of data and knowledge bases, logic and complexity, problem decompositions, and, on the more applied side, web data extraction, and database query processing. Gottlob has received the Wittgenstein Award from the Austrian National Science Fund, is an ACM Fellow, an ECCAI Fellow, a Fellow of the Royal Society, and a member of the Austrian Academy of Sciences, the German National Academy of Sciences, and the Academia Europaea. He chaired the Program Committees of IJCAI 2003 and ACM PODS 2000. He was the main founder of Lixto, a company that provides tools and services for semi-automatic web data extraction which was acquired by McKinsey & Company in 2013. Gottlob was awarded an ERC Advanced Investigator's Grant for the project "DIADEM : Domain-centric Intelligent Automated Data Extraction Methodology". Based on results of this project, he co-founded Wrapidity Ltd, a company that specializes in fully automated web data extraction that was recently acquired by Meltwater, an international media intelligence firm.

Abstract : Reasoning with and about big data, in particular, massive web data is a great challenge. On one hand, we aim for powerful inference mechanisms that add value by creating knowledge from the data. Such mechanisms seem to require sophisticated logics with a high expressive power. On the other hand, we need swift inference algorithms with an acceptable computational complexity. In this talk, reasoning formalisms that achieve both are presented : We introduce and describe specific KRR formalisms for big data that belong to the Datalog+/- family of languages. These logical languages extend the well-known Datalog language by additional features (the "+") to gain expressive power, but simultaneously make syntactic restrictions (the "-") so as to achieve tractability and scalability. After discussing the theoretical foundations of Datalog+/-, some applications to ontological reasoning, web data extraction, data wrangling, and general reasoning about data will be illustrated, among which are some recent commercial applications.

2.3 Galois connections for dependencies in databases
Sergei O. Kuznetsov

Bio : Sergei O. Kuznetsov graduated in 1985 from the Moscow Physical-Technical Institute, Department of Control and Applied Mathematics with Diploma on Combinatorial and Logical Issues of a Plausible Reasoning System. From 1985 to 2006 was a researcher at the All-Russian Institute for Scientific and Technical Information (VINITI) of the Russian Academy of Sciences (Moscow). In 1990 defended "Candidate of Science" (PhD equivalent) thesis "On algorithmic and knowledge representation issues of a machine learning system (JSM-method)" in Theoretical Foundations of Computer Science at VINITI (Moscow). In 2002 defended "Doktor Nauk" (habilitation) thesis "A Theory of Machine Learning in Concept Lattices" at the Computer Center of the Russian Academy of Sciences. In 1999-2004 he was Humboldt fellow and invited professor at the Department of Mathematics and Science, Dresden Technical University (Dresden, Germany). From 2006 he has been Professor of the National Research University Higher School of Economics (HSE), Head of Department of Data Analysis and Artificial Intelligence, Head of International Laboratory for Intelligent Systems and Structural Analysis, HSE (Moscow).

Abstract : Dependencies in databases were an important issue starting from the first works on databases. Functional dependencies, multivalued dependencies, and other type of dependencies were used for database engineering database decomposability, they were also used to define database schemes. Research in data mining and knowledge discovery urged a new wave of interest to dependencies and their approximated versions, however from another point of view : they are being "mined" from databases, not given in advance. In this talk I show that Galois connections, a construction from order and lattice theory, allows for a general view on dependencies, relating them to other tools of knowledge discovery, such as domain taxonomies and biclusters. Algorithmic issues of various problems related to generation and inference of dependencies will be discussed.

3 Tutoriels

3.1 IoT Big Data Stream Mining
Albert Bifet

Bio : Albert Bifet is Associate Professor at Telecom ParisTech. Previously he worked at Huawei Noah's Ark Lab in Hong Kong, Yahoo Labs in Barcelona, University of Waikato and UPC BarcelonaTech. He is the author of a book on Adaptive Stream Mining and Pattern Learning and Mining from Evolving Data Streams. He is one of the leaders of MOA and Apache SAMOA software environments for implementing algorithms and running experiments for online learning from evolving data streams. He was serving as Co-Chair of the Industrial track of IEEE MDM 2016, ECML PKDD 2015, and as Co-Chair of BigMine (2017-2012), and ACM SAC Data Streams Track (2018-2012).

Abstract : The challenge of deriving insights from the Internet of Things (IoT) has been recognized as one of the most exciting and key opportunities for both academia and industry. Advanced analysis of big data streams from sensors and devices is bound to become a key area of data mining research as the number of applications requiring such processing increases. Dealing with the evolution over time of such data streams, i.e., with concepts that drift or change completely, is one of the core issues in IoT stream mining. This tutorial is a gentle introduction to mining IoT big data streams. The first part introduces data stream learners for classification, regression, clustering, and frequent pattern mining. The second part deals with scalability issues inherent in IoT applications, and discusses how to mine data streams on distributed engines such as Spark, Flink, Storm, and Samza.

3.2 Knowledge Graph Expansion and Enrichment
Fatiha Saïs

Bio : Fatiha Saïs is an Associate Professor at Paris Sud University in France. She obtained her Ph.D. in Computer Science at the University of Paris Sud. Her research interest are ontology-based data linking and fusion, RDF data evolution and knowledge discovery from RDF graphs. Her work has been included in several national, industrial and European projects. She has published more than 50 research papers in national and international conferences (AAAI, ISWC, K-Cap) and journals (JWS, KBS and JoDS).

Abstract : Today, we are experiencing an unprecedented production of resources, published as Linked Open Data (LOD, for short). This is leading to the creation of knowledge graphs (KGs) containing billions of RDF (Resource Description Framework) triples, such as DBpedia, YAGO and Wikidata on the academic side, and the Google Knowledge Graph or Microsoft's Satori graph on the commercial side. These KGs contain millions of entities (such as people, proteins, or books), and millions of facts about them. This knowledge is typically expressed in RDF (Resource Description Framework), i.e., as triples of the form ⟨Macron, presidentOf, France⟩. Some KGs provide an ontology expressed in OWL2 (Web Ontology Language), which describes the vocabulary (the classes and properties) for the RDF facts. However, to exploit and take benefits from the richness of this available data and knowledge, several problems have to be faced, namely, data linking, data fusion and knowledge discovery, when data is of big volume, heterogeneous and evolving. In this tutorial we will first give an overview of exiting data linking and key discovery approaches. Then, we will discuss the problem of identity crisis caused by the misuse of owl :sameAs predicate and give some possible solutions. We will finish by highlighting some current challenges in this research area.

3.3 Preference-based Pattern Mining
Bruno Crémilleux, Marc Plantevit, Arnaud Soulet

Bios : Bruno Crémilleux is professor in computer science at the University of Caen-Normandie. He received his PhD in computer science at the University of Grenoble. His main research interests are pattern (set) discovery, Constraint Satisfaction Problems and data mining, preference queries and exploratory data mining.

Marc Plantevit is associate professor in computer sciences at the University of Lyon. He received his PhD in computer science from the University of Montpellier. His research interest include constraint-based pattern mining in general. Currently, he is very interested with sophisticate pattern domains (dynamic/ attributed graphs) and in incorporating background knowledge into pattern mining.

Arnaud Soulet is associate professor in computer science at the University François Rabelais of Tours. He received his PhD at the University of Caen. He has an expertise in constraint-based pattern mining and involvement in the mining process like pattern mining techniques for preference elicitation.

Abstract : This tutorial focuses on the recent shift from constraint-based pattern mining to preference-based pattern mining and interactive pattern mining. Constraint-based pattern mining, which shares common notions with FCA, is now a mature domain of data mining that makes it possible to handle various different pattern domains (e.g., itemsets, sequences, graphs) with a large variety of constraints thanks to solid theoretical foundations and an efficient algorithmic machinery. Even though, it has been realized for a long time that it is difficult for the end-user to model her interest in term of constraints and above to overcome the well-known thresholding issue, researchers have only recently intensified their study of methods for finding high-quality patterns according to the user's preferences.

In this tutorial, we discuss the need of preferences in pattern mining, the principles and methods of the use of preferences in pattern mining. Many methods are derived from constraint-based pattern mining by integrating utility functions or interestingness measures as quantitative preference model. This approach transforms pattern mining in an optimization problem guided by user specified preferences. However, in practice, the user has only a vague idea of what useful patterns could be. The recent research field of interactive pattern mining relies on the automatic acquisition of these preferences and the development of the instant data mining field.

Résumés des articles longs

Optimising SPARQL Query Evaluation in the Presence of ShEx Constraints

Abdullah Abbas
Univ. Grenoble Alpes, CNRS, Grenoble INP, Inria, LIG
Grenoble, France
[firstname.lastname].90@gmail.com

Pierre Genevès
Univ. Grenoble Alpes, CNRS, Grenoble INP, Inria, LIG
Grenoble, France
[firstname.lastname]@cnrs.fr

Cécile Roisin
Univ. Grenoble Alpes, CNRS, Grenoble INP, Inria, LIG
Grenoble, France
[firstname.lastname]@univ-grenoble-alpes.fr.fr

Nabil Layaïda
Univ. Grenoble Alpes, CNRS, Grenoble INP, Inria, LIG
Grenoble, France
[firstname.lastname]@inria.fr

ABSTRACT

ShEx (Shape Expressions) is a language for expressing constraints on RDF graphs. In this work we optimise the evaluation of conjunctive SPARQL queries, on RDF graphs, by taking advantage of ShEx constraints. Our optimisation is based on computing and assigning ranks to query triple patterns, dictating their order of execution. We first define a set of well formed ShEx schemas, that possess interesting characteristics for SPARQL query optimisation. We then define our optimisation method by exploiting information extracted from a ShEx schema. We finally report on evaluation results performed showing the advantages of applying our optimisation on the top of an existing state-of-the-art query evaluation system.

1 INTRODUCTION

The Shape Expressions (ShEx) language [4] allows to describe constraints on RDF graph structures [5]. These descriptions identify predicates and their associated cardinalities and datatypes. ShEx shapes can be used to validate RDF documents, generate RDF documents, or communicate expected graph patterns associated with some process or interface.

In this work we investigate how the evaluation of SPARQL queries [6] can be optimized in the presence of ShEx constraints. We propose a method for optimizing the order of evaluation of subqueries, by taking advantage of the information on the data described in ShEx. While SPARQL query optimisation by static analysis is important and well-studied, the emergence of constraint languages (such as ShEx) raises new questions on how the knowledge of additional constraints can be effectively leveraged as a part of the static analysis and optimization.

In this work, we focus on the logical query structure and in particular on subquery ordering that can be automatically inferred from a set of data constraints. More specifically, we consider SPARQL basic graph patterns (BGPs), and we focus on the order of execution of triple patterns that aim to minimize the overall execution cost of the query.

We postulate that ShEx constraints contain useful information for selecting the order of execution of triple patterns. Optimisation opportunities arise from the presence of joins between query triple patterns, and common variables. In several situations, the order of execution of triple patterns can be rearranged so that the size of intermediate results for join variables are minimized.

We first define a set of well formed ShEx schemas, that possess interesting characteristics for deciding optimal execution orders. We then define our optimisation method by exploiting information extracted from a ShEx schema. Our technique has been implemented on the top of the SPARQLGX [3] query evaluation system and we have shown that it improves the efficiency of query execution times from 5% up to 17%. See [2] for further details.

2 DEFINITIONS

2.1 SPARQL

SPARQL is an RDF query language and a W3C Recommendation.

A SPARQL graph pattern is defined inductively from triple patterns. Given disjoint infinite sets of IRIs - Internationalised Resource Identifiers - (I), blank nodes (B), literals (L), and variables (V), a triple pattern is defined as an instance of $(I \cup B \cup V)(I \cup V)(I \cup B \cup L \cup V)$ denoted by $IBV \times IV \times IBLV$.

In this work we focus on the conjunctive SPARQL fragment which can be defined abstractly as $q ::= t \mid q\ AND\ q'$ where t is a triple pattern.

2.2 ShEx

ShEx (or Shape Expressions) is intended to be an RDF constraint language.

A Shape Expression has a name and describes the constraints associated with a subject RDF node.

Given a finite set of edge labels Σ and a finite set of types Γ, we define a shape expression e over $\Sigma \times \Gamma$ as follows:

$$e ::= \epsilon \mid \Sigma \times \Gamma \mid e^+ \mid (e|e') \mid (e||e')$$

where "|" denotes disjunction, "||" denotes unordered concatenation, and " + " denotes repetition for a positive number of times. From this definition we also further define the following operators as macros:

- $e^? := (\epsilon \mid e)$ (optional)
- $e^* := (\epsilon \mid e^+)$ (unordered Kleene star)

- $e^{[m;n]}$ (e repeated i times with i in the interval from m to n)

which are also parts of the ShEx syntax. In the sequel we write $a :: t$ as a shorthand for $(a, t) \in \Sigma \times \Gamma$.

A shape expression schema (ShEx), or simply a schema, is a tuple $S = (\Sigma, \Gamma, \delta)$, where Σ is a finite set of edge labels, Γ is a finite set of types, and δ is a type definition function that maps elements of Γ to shape expressions e over $\Sigma \times \Gamma$. If δ is not defined for some type $t \in \Gamma$, the default definition is $\delta(t) = \epsilon$.

In this paper, given a shape expression e, a predicate p and a ShEx shape s. we further denote by $(p, s) \in e$ the fact that the restriction (p, s) is necessary in e, and by $(p, s) \in_{opt} e$ that it is optional in e.

3 WELL FORMED DATA-SCHEMA PAIRS

We introduce a notion of well-formed data-schema pairs that will help us to better identify efficient SPARQL query designs, by static analysis of the schema.

The rules for well formation comprise a cardinality rule, that basically tries to avoid the usage of positive and Kleene closures, and au shape distinction rule that puts restrictions on shapes that seem to be too general that they surely miss expressing some constraints that are inherent in the data. These rules guarantee that the necessary information needed for our ranking can be deduced from the ShEx schema. Well-formed data-schema pairs provide the maximal set of desired information that can be inferred from the ranking procedure described in Sect. 5.

4 SHAPE RELATION GRAPH

A shape relation graph is a graphical representation focusing only on the relations existing between shapes in a ShEx document, discarding cardinalities.

Definition 4.1 (Shape Relation Graph). Given a ShEx document S, we define a shape relation graph $G = \mathcal{SRG}(S)$ as a tuple (N, E) of set of nodes N, each corresponding to a ShEx shape, and an labelled directed relation E between nodes such that:

- $E(n_1, x, n_2)$ defines an edge from n_1 to n_2 labeled with x.
- Given any two nodes $n_1, n_2 \in N$, and any predicate p, then $E(n_1, p, n_2)$ if and only if $(p, n_2) \in \delta(n_1)$ and $(p, n_2) \notin_{opt} \delta(n_1)$
- Given any two nodes $n_1, n_2 \in N$, and any predicate p, then $E(n_1, p^{\epsilon}, n_2)$ if and only if $(p, n_2) \in_{opt} \delta(n_1)$

5 RANKING

In order to decide the order of execution of query triple patterns, we assign them ranks inferred from the analysis of the ShEx document. These ranks are based on two main concepts: 1) The hierarchical relations between ShEx shapes, and 2) The predicate distributions among ShEx shapes.

The first concept gives rankings to shapes, and the second concept gives ranking to predicates. The ranking of query triple patterns is based on the product of both rankings together.

5.1 Hierarchical Relations between ShEx Shapes

In ShEx, the definition of a shape may be based on other shapes defined in the same schema. This notion, called shape inclusion, is explicitly represented by the edges of the *shape relation graph* defined in Sect. 4. Such edge relations between shapes allow us to infer information about the relative frequency of data corresponding to these shapes by traversing the *shape relation graph* and annotating relative ranks.

5.2 Predicate Distributions Among ShEx Shapes

Shape ranking is not sufficient for deciding rankings of triple patterns in a query since such ranking is also affected by the uniqueness versus globality of predicates within shapes.

This step works by reducing the ranking of a predicate when it is more global, i.e. when it is used with more shapes.

5.3 SPARQL Query Triple Rankings

Now our purpose in the final step is to rank the triple patterns given a BGP query. Triple patterns with higher ranking will be executed first. Before ranking triples, we need to validate the BGP against the ShEx document, and for each subject in the triple patterns the ShEx validator will decide to which shapes this subject may belong. A subject may belong to multiple shapes at the same time, thus we use averaging among candidate shapes for each of the triple patterns.

6 CONCLUSION

We studied a method for SPARQL query optimisation based on ranking triple patterns in order to select their execution order. The originality of our approach is that rankings generated by our system are based on information inferred from a schema expressed in ShEx, which is an emerging schema language for RDF data. To the best of our knowledge, this is the first attempt of leveraging ShEx constraints for SPARQL query optimization.

Preliminary experimental results (as it has been shown in [2]) indicate that most rankings found by our system lead to improvements in query execution times. This illustrates the interest of considering ShEx constraints for SPARQL query optimisation.

This work takes part of a PhD thesis done by the main author with more elaborations on the topic [1].

REFERENCES

[1] Abdullah Abbas. 2017. *Static Analysis of Semantic Web Queries with ShEx Schema Constraints.* Ph.D. Dissertation. http://www.theses.fr/2017GREAM064 Directed by Nabil Layaïda, Pierre Genevès, and Cécile Roisin. Université Grenoble Alpes 2017.

[2] Abdullah Abbas, Pierre Genevès, Cécile Roisin, and Nabil Layaïda. 2018. Selectivity Estimation for SPARQL Triple Patterns with Shape Expressions. In *ICWE'18 - 18th International Conference on Web Engineering.* Springer, Cáceres, Spain, 195–209. https://doi.org/10.1007/978-3-319-91662-0_15

[3] Damien Graux, Louis Jachiet, Pierre Genevès, and Nabil Layaïda. 2016. *SPARQLGX: Efficient Distributed Evaluation of SPARQL with Apache Spark.* Springer International Publishing, Cham, 80–87. https://doi.org/10.1007/978-3-319-46547-0_9

[4] Eric Prud'hommeaux, Jose Emilio Labra Gayo, and Harold Solbrig. 2014. Shape Expressions: An RDF Validation and Transformation Language. In *Proceedings of the 10th International Conference on Semantic Systems (SEM '14).* ACM, New York, NY, USA, 32–40. https://doi.org/10.1145/2660517.2660523

[5] Guus Schreiber and Yves Raimond. 2014. *RDF 1.1 Primer.* W3C Note. W3C. http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/.

[6] Andy Seaborne and Steven Harris. 2013. *SPARQL 1.1 Query Language.* W3C Recommendation. W3C. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

Continuous processing of diversity-aware top-k queries in social networks

Abdulhafiz Alkhouli
Dan Vodislav

ABSTRACT

A great part of the Web content is produced and consumed today as information streams, especially in the context of social networks. Efficient processing of ranking queries over such streams at the social network scale requires continuous processing, but becomes very challenging when the relevance model combines content, time and social network criteria. We consider here the problem of adding diversity requirements for the results of continuous top-k queries in such a large scale social network context, while preserving an efficient, continuous processing. We propose the DA-SANTA algorithm, which smoothly adds content diversity to the continuous processing of top-k queries at the social network scale. A rich experimental study demonstrates the very good properties in terms of effectiveness and efficiency of this algorithm.

Une approche par circuit pour une énumération efficace

Antoine Amarilli
Pierre Bourhis
Louis Jachiet
Stefan Mengel

ABSTRACT

Nous étudions le problème d'énumérer les valuations satisfaisant un circuit tout en limitant le délai, c'est-à-dire le temps nécessaire pour calculer chaque valuation successive. Nous nous concentrons sur la classe des circuits d-DNNF initialement introduits pour la compilation des connaissances, un sous-domaine de l'intelligence artificielle. Nous proposons un algorithme pour ces circuits qui énumère les valuations avec un prétraitement linéaire et le délai linéaire dans la taille de chaque valuation. En outre, les valuations ayant un poids de Hamming constant peuvent être énumérées avec un prétraitement linéaire et un délai constant. Notre cadre d'énumération efficace s'applique ainsi à tous les problèmes dont les solutions peuvent être compilées dans une d-DNNF. En particulier, nous l'utilisons pour redémontrer des résultats classiques dans la théorie des bases de données, pour les bases de données factorisées et pour l'évaluation MSO. Plus précisément, nous donnons une preuve indépendante de l'énumération à délai constant pour les formules MSO avec des variables libres de premier ordre sur les structures de largeur d'arbres bornéesWe study the problem of enumerating the satisfying valuations of a circuit while bounding the delay, i.e., the time needed to compute each successive valuation. We focus on the class of structured d-DNNF circuits originally introduced in knowledge compilation, a sub-area of artificial intelligence. We propose an algorithm for these circuits that enumerates valuations with linear preprocessing and delay linear in the Hamming weight of each valuation. Moreover, valuations of constant Hamming weight can be enumerated with linear preprocessing and constant delay. Our results yield a framework for efficient enumeration that applies to all problems whose solutions can be compiled to structured d-DNNFs. In particular, we use it to recapture classical results in database theory, for factorized database representations and for MSO evaluation. This gives an independent proof of constant-delay enumeration for MSO formulae with first-order free variables on bounded-treewidth structures

Counting Types for Massive JSON Datasets

Mohamed-Amine Baazizi
Dario Colazzo
Giorgio Ghelli
Carlo Sartiani

ABSTRACT

Type systems express structural information about data, are human readable and hence crucial for understanding code, and are endowed with a formal definition that makes them a fundamental tool when proving program properties. Internal data structures of a database store quantitative information about data, information that is essential for optimization purposes, but is not used for documentation or for correctness proofs.In this paper we propose a new idea: raising a part of the quantitative information from the system-levelstructures to the type level.Our proposal is motivated by the problem of schema inference for massive collections of JSON data,which are nowadays often collected from external sources and stored in NoSQL systemswithout an a-priori schema, which makes a-posteriori schema inference extremely useful.NoSQL systems are oriented towards the management of heterogeneous data, and in this context we claim that quantitative information is important in order to assess the relative weight of different variants.We propose a type system

Optimisation du Temps de Communication via la Configuration du Middleware

Abdeslem Belghoul, Mourad Baïou, Radu Ciucanu, Farouk Toumani
Université Clermont Auvergne & CNRS LIMOS, France
{belghoul, baiou, ciucanu, ftoumani}@isima.fr

ABSTRACT

Minimiser le temps de communication des résultats de requêtes sur le réseau présente un défi fondamental dans les systèmes de gestion de bases de données distribuées. Dans cet article, nous abordons le problème d'optimisation du temps de communication de données dans les systèmes de gestion de données distribuées, en focalisant sur la relation entre le temps de communication de données et la configuration du middleware (tel que JDBC, ODBC ou propriétaire). Nous focalisons sur deux paramètres du middleware qui sont adaptés manuellement par les administrateurs de BD ou les programmeurs : la *taille du batch* F (le nombre de tuples communiqués à la fois) et la *taille du message* M (la taille du buffer au niveau middleware). Nous avons réalisé une étude expérimentale mettant l'accent sur l'impact crucial des paramètres F et M sur le temps de communication de données. De plus, nous proposons le framework $MIND$ qui ajuste les dits paramètres tout en s'adaptant aux différentes requêtes et aux différents environnements réseaux. Les principales contributions techniques de $MIND$ sont (i) une *fonction d'estimation du temps de communication* qui prend en compte les paramètres F et M, la taille du résultat et l'environnement réseau, et (ii) un *algorithme itératif d'optimisation* pour trouver les valeurs de F et M qui minimisent le temps de communication et consomment moins de ressources. Nous concluons avec une étude expérimentale mettant l'accent sur l'efficacité de $MIND$.

1 INTRODUCTION

Le transfert de données sur un réseau de communication de données est une tâche inhérente au traitement de requêtes distribuées dans les diverses architectures de gestion de données distribuées [6, 8] (par exemple, systèmes client-serveur, point à point, parallèle et d'intégration de données (médiation)). Dans toutes ces architectures, un nœud donné (jouant le rôle de serveur, de client, de médiateur, etc.) peut envoyer une requête (ou une sous-requête) à un autre nœud (un serveur ou un médiateur) qui va exécuter la requête et renvoyer les résultats de la requête au nœud demandeur.

Malgré les énormes progrès réalisés dans les technologies réseaux et de télécommunications, d'une part, et les techniques de traitement réparti et de gestion des données, d'autre part, le coût du transfert de données (appelé aussi temps de communication) reste souvent une source importante de problèmes de performances. Cela est dû à la charge croissante imposée par les applications modernes à forte intensité de données.

2 CONTEXTE ET MOTIVATION

Dans cet article, nous présentons le problème d'optimisation du temps de transfert de données dans les systèmes de gestion de données distribuées, en focalisant sur la relation entre le temps de communication de données et la configuration du middleware. En réalité, le middleware détermine, entre autres, comment les données sont divisées en lots de F tuples et messages de M octets avant d'être communiqués à travers le réseau. Concrètement, nous nous concentrons notre travail sur la question de recherche suivante : étant donnée une requête Q et l'environnement réseau, quelle est la meilleure configuration de F et M qui minimisent le temps de communication du résultat de la requête à travers le réseau ?

A notre connaissance, ce problème n'a jamais été étudié par la communauté de recherche en base de données.

3 TRAVAUX LIÉS

L'état de l'art dans le domaine de l'optimisation du coût de communication de données dans le traitement de requêtes distribuées fait ressortir plusieurs travaux de recherche initiés dans ce domaine et qui se sont focalisés principalement sur l'élaboration de plans distribués de requêtes qui minimisent le coût de communication de données [1, 4–8].

Notre travail est complémentaire à ceux cités dans la précédente paragraphe parce que nous nous concentrons sur la façon dont le résultat de requête est communiqué sur le réseau, plus précisément sur la façon d'adapter les paramètres du middleware afin de réduire le temps de communication du résultat de requête [2, 3].

4 CONTRIBUTIONS

Les principales contributions de notre travail de recherche sont [2, 3] :

- une étude expérimentale qui met en évidence l'impact de la configuration du middleware sur le temps de transfert de données. Dans cette étude, nous explorons deux paramètres du middleware que nous avons empiriquement identifié comme ayant une influence importante sur le coût de transfert de données :

(i) la taille du lot F (c'est-à-dire le nombre de tuples dans un lot qui est communiqué à la fois vers une application consommant des données) et (ii) la taille du message M (c'est-à-dire la taille en octets du tampon du middleware qui correspond à la quantité de données à transférer à partir du middleware vers la couche réseau).

En effet, l'adaptation de ces paramètres est dépendante aux différentes requêtes (dont la sélectivité varie) et aux environnements réseaux (qui varient en termes de débit).

- Ensuite, nous décrivons un modèle de coût permettant d'estimer le temps de transfert de données. Ce modèle de coût est basé sur la manière dont les données sont transférées entre les nœuds de traitement de données. Notre modèle de coût est basé sur deux observations cruciales : (i) les lots et les messages de données sont communiqués différemment sur le réseau : les lots sont communiqués de façon synchrone et les messages dans un lot sont communiqués en pipeline (asynchrone) et (ii) en raison de la latence réseau, le coût de transfert du premier message d'un lot est plus élevé que le coût de transfert des autres messages du même lot.
- Nous proposons une stratégie pour calibrer les poids du premier et non premier messages dans un lot. Ces poids sont des paramètres dépendant de l'environnement réseau et sont utilisés par la fonction d'estimation du coût de communication de données.
- Enfin, nous développons un algorithme d'optimisation permettant de calculer les valeurs des paramètres F et M qui fournissent un bon compromis entre un temps optimisé de communication de données et une consommation minimale de ressources. L'approche proposée a été validée expérimentalement en utilisant des données issues d'une application en Astronomie.

5 CONCLUSIONS

Dans notre travail, nous avons montré que la configuration du middleware avait un impact majeur sur le temps de communication d'un résultat de requête dans un environnement distribué. Nous avons ensuite présenté le cadre $MIND$, qui définit deux paramètres de middleware (la taille de lot F et la taille de message M), tout en s'adaptant aux différentes requêtes (dont la sélectivité varie) et aux environnements réseaux (qui varient en termes de débit). Les principales contributions techniques de $MIND$ sont une fonction d'estimation du temps de communication (prenant en compte les paramètres du middleware, la taille du résultat de la requête et l'environnement du réseau) et un algorithme d'optimisation itératif (permettant un bon compromis entre une faible consommation de ressources et faible coût de communication).

Dans la perspective des travaux futurs, de nombreuses pistes d'investigation sont possibles, par exemple l'intégration de $MIND$ dans l'optimiseur de requêtes pour l'adaptation du middleware et l'élaboration de plans de requêtes distribuées tenant compte l'adaptation du middleware.

REFERENCES

[1] Paul Beame, Paraschos Koutris, and Dan Suciu. 2013. Communication steps for parallel query processing. In *PODS*. 273–284.
[2] A. Belghoul. 2017. *Optimizing communication cost in distributed query processing.*
[3] A. Belghoul, M. Baïou, and F. Toumani. 2018. MIND: An approach to optimize communication time via middleware tuning. In *Information Systems*.
[4] Sumit Ganguly, Akshay Goel, and Abraham Silberschatz. 1996. Efficient and Acurate Cost Models for Parallel Query Optimization. In *PODS*. 172–181.
[5] Laura M. Haas, Donald Kossmann, Edward L. Wimmers, and Jun Yang. 1997. Optimizing Queries Across Diverse Data Sources. In *VLDB*. 276–285.
[6] Donald Kossmann. 2000. The State of the art in distributed query processing. *ACM Comput. Surv.* 32, 4 (2000), 422–469.
[7] Lothar F. Mackert and Guy M. Lohman. 1986. R* Optimizer Validation and Performance Evaluation for Distributed Queries. In *VLDB*. 149–159.
[8] M. Tamer Özsu and Patrick Valduriez. 2011. *Principles of Distributed Database Systems, Third Edition.* Springer.

JSON: Modèle de données, langage de requête et de schéma

Pierre Bourhis
Juan Reuters
Fernando Suárez
Domagoj Vrgoč

ABSTRACT

Malgré le fait que JSON est actuellement l'un des formats les plus populaires pour l'échange de données sur le Web, il existe très peu d'études sur ce sujet et il n'y a pas d'accord sur un cadre théorique pour représenter la structure de données représentée par JSON. Par conséquent, dans cet article, nous proposons un modèle de données formel pour les documents JSON et, en fonction des fonctionnalités communes présentes dans les systèmes disponibles utilisant JSON, nous définissons un langage de requêtes léger qui nous permet de naviguer à travers les documents JSON. Nous introduisons également une logique pour définir des schémas sur JSON et étudions la complexité des problèmes classiques associées à ces deux formalismes.Despite the fact that JSON is currently one of the most popular formats for exchanging data on the Web, there are very few studies on this topic and there are no agreement upon theoretical framework for dealing with JSON. There- fore in this paper we propose a formal data model for JSON documents and, based on the common features present in available systems using JSON, we define a lightweight query language allowing us to navigate through JSON documents. We also introduce a logic capturing the schema proposal for JSON and study the complexity of basic computational tasks associated with these two formalisms

Large Scale Density-friendly Graph Decomposition via Convex Programming

Maximilien Danisch
Hubert Chan
Mauro Sozio

ABSTRACT

Algorithms for finding dense regions in an input graph have proved to be effective tools in graph mining and data analysis. Recently, Tatti and Gionis [WWW 2015] presented a novel graph decomposition (known as the locally-dense decomposition) that is similar to the well-known k-core decomposition, with the additional property that its components are arranged in order of their densities. Such a decomposition provides a valuable tool in graph mining. Unfortunately, their algorithm for computing the exact decomposition is based on a maximum-flow algorithm which cannot scale to massive graphs, while the approximate decomposition defined by the same authors misses several interesting properties. This calls for scalable algorithms for computing such a decomposition. In our work, we devise an efficient algorithm which is able to compute exact locally-dense decompositions in real-world graphs containing up to billions of edges. Moreover, we provide a new definition of approximate locally-dense decomposition which retains most of the properties of an exact decomposition, for which we devise an algorithm that can scale to real-world graphs containing up to tens of billions of edges. Our algorithm is based on the classic Frank-Wolfe algorithm which is similar to gradient descent and can be efficiently implemented in most of the modern architectures dealing with massive graphs. We provide a rigorous study of our algorithms and their convergence rates. We conduct an extensive experimental evaluation on multi-core architectures showing that our algorithms converge much faster in practice than their worst-case analysis. Our algorithm is even more efficient for the more specialized problem of computing a densest subgraph.

PSH-DB, un système clé-valeur permettant l'indexation et la recherche de séquences ADN

Jocelyn DE GOËR DE HERVE*
jocelyn.degoer@inra.fr
EPIA, INRA, VetAgro Sup
63122 Saint Genès Champanelle,, France
LIMOS Université Clermont Auvergne, CNRS
F-63000 Clermont-Ferrand, France

Myoung-Ah KANG*
kang@isima.fr
LIMOS Université Clermont Auvergne, CNRS
F-63000 Clermont-Ferrand, France

Xavier BAILLY*
xavier.bailly@inra.fr
EPIA, INRA, VetAgro Sup
63122 Saint Genès Champanelle,, France

Engelbert MEPHU-NGUIFO*
engelbert.mephu_nguifo@uca.fr
LIMOS Université Clermont Auvergne, CNRS
F-63000 Clermont-Ferrand, France

ABSTRACT

L'évolution constante des techniques de séquençage de l'ADN, entraîne la production de plus en plus massive de données génomiques, pour un coût de plus en plus bas. L'apparition de séquenceurs toujours plus modulables et portatifs, qui permettent la lecture de fragments d'ADN de plus en plus long, va conduire à une démocratisation certaine de ces outils. Le stockage et le traitement de cette masse de données en constante évolution, reste un enjeu majeur pour les années à venir. Durant le processus d'analyse des données génomiques, la recherche de sous-séquences, au travers de bases de données de génomes de référence, est une tâche incontournable. Compte tenu de l'accroissement des données à analyser, l'un des enjeux est de créer des algorithmes permettant d'identifier rapidement les génomes de référence les plus proches, d'une séquence nouvellement séquencée. Ceci afin d'effectuer un rapide prétraitement permettant de conserver uniquement des séquences de références pertinentes pour qu'elles soient par la suite analysées par des outils plus spécifiques aux questionnements biologiques.

Nous présentons PSH-DB (Perceptual Sequence Hashing DataBase), un système clé-valeur in-memory, utilisant la fonction de hachage perceptuel PSH (Perceptual Sequence Hashing), que nous avons conçue spécifiquement pour l'indexation des séquences nucléotidiques de type ADN ou ARN. Cette première implémentation a pour objectif de démontrer l'intérêt de la fonction PSH, pour indexer et rechercher des séquences exactes ou proches au sein d'une base de données de séquences nucléotidiques de référence. La caractéristique principale du moteur PSH-DB est qu'il permet de stocker nativement les clés de hachage sous forme de chaˆbinaires (bitset), contrairement à la majorité des moteurs NoSQL, qui stockent les clés sous forme de nombres entiers ou de chaines de caractères. Le stockage au format bitset permet d'optimiser l'espace de stockage, mais surtout de calculer nativement et rapidement des distances de Hamming, entre deux clés de hachage, via l'utilisation de l'opérateur booléen XOR et de l'instruction de bas niveau POPCOUNT. Lors de la phase d'indexation, les clés de hachage sont calculées à partir des séquences ADN fournies en entrée et stockées sous la forme de chaˆbinaires. Ceci permet nativement le calcul de Distance de Hamming, permettant la comparaison des clés de hachage et ainsi à partir d'une séquence requête, de retourner les séquences ADN les plus proches, présentes dans la base de données.

La fonction de hachage PSH, est basée sur des concepts de hachage perceptuel, utilisés habituellement pour indexer et comparer des images numériques. Les séquences ADN/ARN à indexer, doivent être converties sous la forme de matrices de pixels, afin de créer de nouvelles structures 2-D directement représentatives de celles-ci. Les clés de hachage sont générées à l'aide d'une fonction Transformée en Cosinus Discrète à Coefficients Signés. Elles s'avèrent être moins volumineuses que les séquences à partir desquelles elles ont été calculées. Malgré une diminution importante de la taille des données entre une séquence et sa clé de hachage correspondante, les clés de hachage conservent la propriété de pouvoir être comparées à l'aide de la Distance de Hamming. Par conséquent, deux clés de hachage générées à partir de deux séquences proches, auront une distance de Hamming faible.

Les expérimentations menées à partir de données réelles démontrent les capacités d'indexation et de recherche du système PSH-DB, en termes de sensibilité par rapport à BLAST, l'outil de référence en bio-informatique, mais aussi en termes de diminution des données et de vitesse d'exécution des fonctions de recherche par rapport au système clé-valeur REDIS.

CCS CONCEPTS

• **Theory of computation → Bloom filters and hashing**; • **Applied computing → Bioinformatics**.

KEYWORDS

Séquence ADN, hachage perceptuel, table de hachage, indexation

Apprentissage de points communs entre requêtes SPARQL

Sara El Hassad
Univ Rennes, CNRS, IRISA
Lannion, France
sara.el-hassad@irisa.fr

François Goasdoué
Univ Rennes, CNRS, IRISA
Lannion, France
fg@irisa.fr

Hélène Jaudoin
Univ Rennes, CNRS, IRISA
Lannion, France
helene.jaudoin@irisa.fr

ABSTRACT

La recherche de points communs entre des descriptions de données ou de connaissances est un problème de raisonnement fondamental en Apprentissage Automatique, qui a été formalisé dans les années 70 sous la forme du calcul de *plus petits généralisants (*lgg*)* de ces descriptions.

Nous revisitons ce problème dans le cadre du langage de requête SPARQL pour les graphes RDF. Contrairement à l'état de l'art, nous traitons le problème pour *toute* la classe des requêtes conjunctives SPARQL, connues sous le nom de Basic Graph Pattern Queries. Par ailleurs, quand des *connaissances du domaine* sont disponibles sous forme de contraintes ontologiques de RDF Schema, nous tirons profit de celles-ci pour exhiber des lggs beaucoup plus précis, comme l'attestent les expérimentations menées sur le jeu de données DBpedia.

KEYWORDS

Basic Graph Pattern queries· RDF · RDFS · plus petit généralisant

ACKNOWLEDGMENT

Ce travail a été partiellement financé par Lannion-Tregor Communauté et la Région Bretagne (projet PAWS).

Schema Mappings for Data Graphs

Nadime Francis
Leonid Libkin

ABSTRACT

Les correspondances de schémas sont un concept fondamental dans le cadre de l'échange et de l'intégration de données, qui ont été abondamment étudiées pour divers modèles de données. Dans le cas des graphes de données, ces correspondances n'ont été étudiées que dans un scénario assez éloigné de la pratique, où l'on s'intéresse uniquement à la topologie des graphes, et non aux données qu'ils contiennent. Nous nous intéressons en particulier au problème du calcul de la réponse certaine, dans les contextes de l'échange et de l'intégration de données, et ce pour des modèles de graphes de données alliant topologie et données individuelles. Notre contribution principale montre que le fait de pouvoir interroger ces données individuelles conduit à des résultats s'écartant largement des cas déjà étudiés.Le modèle de graphes de données que nous considérons ici est une abstraction théorique proche des graphes de propriété utilisés dans les systèmes réels de gestion de données structurées sous la forme de graphes. Notre premier résultat est très fortement négatif : dès lors que la correspondance peut exprimer des propriétés navigationnelles même très simples, le problème de calculer la réponse certaine pour des requêtes interrogeant simultanément la topologie et les données du graphe devient indécidable. Ce résultat montre qu'il est nécessaire, au moins dans le cadre de l'intégration et de l'échange de données, de se restreindre à des correspondances ne spécifiant aucune forme de clôture transitive sur la structure du graphe cible. Le calcul de la réponse certaine redevient alors de nouveau décidable pour de telles correspondances et des extensions des requêtes régulières manipulant également les données individuelles du graphe, mais demeure intractable. Nous proposons alors une méthode permettant d'obtenir des approximations efficaces du résultat sans limiter davantage le pouvoir d'expression des langages de requêtes mis en jeu. Cette méthode s'appuie sur une représentation des valeurs manquantes (nulls) similaire à celle utilisée dans les systèmes de gestion de données traditionnels, plutôt que d'utiliser des valeurs marquées (marked nulls) comme il est d'usage dans les scénarios d'échange et d'intégration de données.

Clustering collaboratif : Principes et mise en œuvre

Pierre Gançarski
Antoine Cornuéjols
Cédric Wemmert
Younès Bennani

ABSTRACT

Pour tenter de faire sens des masses de données disponibles en quantité croissante, il est nécessaire de disposer d'outils performants limitant l'implication, souvent chronophage, de l'expert. Les méthodes non supervisées d'exploration de données telles que les méthodes de clustering sont une réponse à ce besoin. Cependant, leur mise en oeuvre effective demande que l'utilisateur présuppose un certain nombre de propriétés des structures internes des données à mettre en évidence telles que par exemple leur type ou simplement leur nombre. Il doit aussi être capable de traduire cet a priori sous forme d'un ou plusieurs objectifs d'analyse et de choisir les méthodes adéquates (et leurs paramètres) de façon optimale. Malheureusement, ceci est une tâche ardue pour laquelle il n'existe pas actuellement de "recette miracle". Par ailleurs, l'utilisation de bases de données distantes nécessite des analyses locales préservant la confidentialité, pour lesquelles il peut être intéressant de partager des informations.Dans ce contexte, les approches collaboratives, dans lesquelles des algorithmes de clustering, pouvant être paramétrés différemment, travaillant sur des données éventuellement différentes échangent et partagent des informations apparaissent comme une voie prometteuse. En effet, les expériences montrent que l'influence des choix initiaux des méthodes et de leur paramètres sont fortement atténués et qu'un tel schéma permet de découvrir de meilleures structures que si chacune de ces méthodes travaillait isolément.L'objectif de cet article est de présenter dans un premier temps, les questions que posent le clustering en général, et le clustering distribué en particulier ainsi que les défis à relever. Dans un deuxième temps, il s'agit d'établir un cadre dans lequel il est possible d'organiser et d'inscrire les approches possibles de clustering collaboratif. Les différentes possibilités sont illustrées par des exemples de travaux existants.

Une classification expérimentale multi-critères des évaluateurs SPARQL répartis

Damien Graux
Louis Jachiet
Pierre Geneves
Nabil Layaida

ABSTRACT

SPARQL est le langage standard pour interroger des données au format RDF. Il exite une grande variété d'évaluateurs SPARQL mettant en place différentes architectures tant pour la répartition des données que pour le déroulement des calculs. Ces différences couplées à des optimisations spécifiques pour chaque évaluateurs, par exemple le pré-traitement des données ou leur indexation, rendent la comparaison entre ces systèmes impossible d'un point de vue théorique. Ainsi, de nombreux systèmes traitant de l'évaluation de requêtes SPARQL ont été développés, chacun mettant en place une stratégie particulière adaptée à un context précis. Nous proposons un nouvel angle de comparaison des évaluateurs SPARQL répartis basé sur un classement multi-critères. Nous suggérons d'utiliser un ensemble de cinq fonctionnalités (nommées vélocité, immédiateté, dynamicité, parcimonie et robustesse) afin d'obtenir une description plus fine des comportements des evaluateurs répartis plutôt que de considérer l'analyse plus traditionnelle des performances temporelles. Nous montrons comment de telles fonctionnalités aident à savoir plus précisément dans quels cas d'utilisation un système donné sera plus souhaitable qu'un autre. Afin d'illustrer cette méthode, nous avons mené des expérimentations mettant en compétition dix systèmes existants que nous avons ensuite classés en utilisant une grille de lecture aidant à la visualisation des avantages et des limitations des techniques dans le domaine de l'évaluation répartie de requêtes SPARQL.

ALGeoSPF: A Hierarchical Geographical Factorization Model for POI Recommendation

Jean-Benoît Griesner
LTCI, Télécom ParisTech
Paris, France
griesner@telecom-paristech.fr

Talel Abdessalem
LTCI, Télécom ParisTech
UMI CNRS IPAL
National University of Singapore
Paris, France
talel.abdessalem@telecom-paristech.fr

Hubert Naacke
UPMC Université Paris 06, LIP6
Paris, France
Hubert.Naacke@lip6.fr

Pierre Dosne
LTCI, Télécom ParisTech
Paris, France
pierre.dosne@telecom-paristech.fr

ABSTRACT

The task of points-of-interest (POI) recommendations has become an essential feature in location-based social networks (LBSNs) with the significant growth of shared data on LBSNs. However it remains a challenging problem, because the decision process of a user choosing to visit a POI depends on numerous factors. The high level of sparsity of the data in LBSNs makes the POI recommendation problem even more challenging, especially for large geographical areas and worldwide datasets. Moreover, in this context the mobility behavior of the users is very heterogeneous, ranging from urban to worldwide mobility.

In this paper, we explore the impact of spatial clustering on the recommendation quality. The proposed approach combines spatial clustering with users' influences. It is based on a Poisson factorization model built on an implicit social network, inferred from the geographical mobility patterns. We conduct a comprehensive performance evaluation of our approach on the YFCC dataset (a very large-scale real-world dataset). The experiments show that our approach achieves a significantly superior recommendation quality compared to other state-of-the-art recommendation techniques.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; *Learning to rank*; *Search engine indexing*;

KEYWORDS

Recommendation, Poisson, Factorization, POI, Accessibility

1 INTRODUCTION

Personalized POI recommendation is the task of proposing to a user a list of relevant POI the user could be interested to visit. This task has become an essential component of LBSN. Through these networks millions of users can share their experiences and their comments concerning the POI that they have visited in the past. These visits are also known as *check-in* activities that correspond to users' preferences on POI. Dealing with LBSN check-ins involves to take into account of several challenging characteristics of POI recommendation: (i) high level of **sparsity** which means that the density of the user-POI check-in matrix is very low in LBSN [2–4, 9, 10], (ii) **frequency data**: we know only how many times a user has been located to a place. Most of existing works use Gaussian assumption to model the geographical users' mobility [2, 5, 8], (iii) **implicit feedback**: we know which POI have been checked in but we cannot know if the user's experiences have been positive or negative. In this case, there is no straightforward way to distinguish between unattractive POI for the user and those undiscovered by the user but potentially attractive for her, and finally (iv) **contextual information**: adding contextual information into the model requires to divide the user-POI check-in matrix into a tensor, which increases even more the sparsity.

Contributions: In this work we propose an efficient factorization model for POI recommendation (namely ALGeoSPF) that takes into account the contextual information as well as the social influence based on specific users' mobility behaviors. Moreover we exploit a flexible hierarchical clustering structure to detect these behaviors. We can summarize the contributions we achieve in this paper as follows:

- We propose a **scalable** probabilistic factorization approach for POI recommendation problem.
- We introduce the concept of **accessibility** between two different POI to model the probability for a user to move from one POI to another.
- We propose a **hierarchical structure** to define several levels of **superPOI** thanks to a flexible clustering algorithm.
- We build more **personalized recommendations** based on users specific mobility behaviors.
- We build a location-location accessibility-aware directed graph (**AGRA**) and propose numerous ways to derive from it implicit information that we integrate into a social Poisson factorization method.
- Finally we conduct exhaustive experiments on a large-scale dataset which confirm the **efficiency** of our approach.

2 GENERAL IDEA

GeoSPF exploits the assumption that it exists a combination of geographical and social influences, and personal preferences behind the decision process of the user. If we denote by $\alpha(u,p)$ the degree of interest of a user u has for a POI p, $\mathcal{S}(u,p)$ the social influence user u got on p, and $\mathcal{G}(u,p)$ the geographical preference of user u regarding p, the probability to observe the pair (u,p) in the dataset should be proportional to the interest of u for p:

$$\mathrm{P}(u,p) \propto \mathbb{F}[\alpha(u,p), \mathcal{G}(u,p), \mathcal{S}(u,p)] \quad (1)$$

where $\mathbb{F}[\cdot]$ is a monotically decreasing function. Based on this general idea the main steps of GeoSPF are as follows: **(i)** build an accessibility-aware graph (AGRA) based on the observed transitions and their probabilities, **(ii)** infer an implicit social network from AGRA and the similarities between the check-in history of the users, and finally **(iii)** integrate the graph into a social Poisson factorization recommendation model.

Our Augmented Local-Global GeoSPF model (denoted ALGeoSPF) consists in defining local and global layers of *superPOI* in order to increase the dataset density. We adopt a clustering approach that consists in dividing the initial geographical space into even rectangular cells. As in [1, 7] a cell can be recursively divided into 4 cells. Thus we construct a tree where the root is the whole world map and each node is a quarter of its parent region. The principle of the algorithm is to recursively divide a cell c until it satisfies the condition concerning the number of different users who made check-ins in that cell: $N(c)<N_{max}$. The algorithm is presented in algorithm 1.

Algorithm 1 Top-down Clustering Method for ALGeoSPF

1: **Input**:
   - N_{max}: maximum number of users having visited a cell.
2: **Global Output**:
   - S: the set of superPOIs cells.
3: **Initialize**: $S \leftarrow \emptyset$
4: **function** WORLDTOSUPERPOIS (C: a cell)
5: Split C into 4 even rectangular cells $C_1,\cdots,C_4$
6: **for each** C_i **do**
7: **if** $N(C_i)>N_{max}$ and $\#POIs(C_i)\geqslant 2$ **then**
8: worldToSuperPOIs (C_i)
9: **else** Put C_i into S

3 RESULTS

We have conducted exhaustive experiments on real-world LBSN datasets. Our experiments use specifically the recall@N and NDCG measure. As expected NMF and PMF do not yield a good quality since they were not designed to cope with implicit feedback datasets. This is consistent with the results in [6]. Although SLIM is known to perform well on sparse datasets, it fails to achieve a good quality in our context because it assumes explicit feedback (instead of implicit one). Among all the state-of-the-art competitors, PF achieves the best quality. Thus, we focus our analysis on comparing PF vs. GeoSPF. As a major result, the relative benefit of GeoSPF on all the datasets is around 200%. This impressive gain makes GeoSPF suitable for POI recommendation over wide geographical areas. It confirms that exploiting restricted contextual information (only GPS and check-in date) through a combined geographical/social solution yields a high quality for POI recommendation.

We also aim to assess the benefit of our geographical clustering approach for user-class aware recommendation. This is why we have analysed the recommendation quality of ALGeoSPF applied on the YFCC dataset considering the urban users isolated from the globetrotters. For every size of the social network, we observe that ALGeoSPF always improves significantly the recall by more than 50%. We can see that ALGeoSPF outperforms the other methods, although BPR yields the closest quality. We observe also that globally the recall measures of tested models for the YFCC dataset are much lower than other datasets: this is due to the low density because of the geographical area covered is large.

4 CONCLUSIONS

In this paper we have proposed a new scalable approach for the POIs recommendation task in LBSNs called ALGeoSPF. The main goal of ALGeoSPF was to build a model which does not suffer from the low density of large-scale geographical datasets, and which takes into account of the specific users' mobility behaviors. Based on the new concepts of *superPOI* and *accessibility* that we have introduced in this work, our approach succeeded **(i)** to build efficiently an implicit scalable factorization model and **(ii)** to capture the user's mobility preferences into a hierarchical structure and finally **(iii)** to present significant better results than baselines on large-scale datasets. We have demonstrated with extensive experiments that ALGeoSPF significantly outperforms all the alternative approaches in terms of *recall* and *NDCG*. To the best of our knowledge, we are among the first to test a POIs recommendation approach on the YFCC dataset for our experiments.

REFERENCES

[1] Marie Al-Ghossein and Talel Abdessalem. 2016. SoMap: Dynamic Clustering and Ranking of Geotagged Posts *(WWW '16 Companion)*. 151–154.
[2] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks *(AAAI, 2012)*.
[3] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring Temporal Effects for Location Recommendation on Location-based Social Networks *(RecSys '13)*.
[4] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: Joint Geographical Modeling and Matrix Factorization for Point-of-interest Recommendation *(KDD '14)*.
[5] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. 2013. Learning Geographical Preferences for Point-of-interest Recommendation *(KDD '13)*.
[6] Bin Liu and Hui Xiong. 2013. Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness *(ICDM'13)*.
[7] Wei Wang, Jiong Yang, and Richard R. Muntz. 1997. STING: A Statistical Information Grid Approach to Spatial Data Mining *(VLDB '97)*. 186–195.
[8] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting Geographical Influence for Collaborative Point-of-interest Recommendation *(SIGIR '11)*.
[9] Jia-Dong Zhang and Chi-Yin Chow. 2013. iGSLR: Personalized Geo-social Location Recommendation: A Kernel Density Estimation Approach *(SIGSPATIAL'13)*.
[10] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Large-Scale Parallel Collaborative Filtering for the Netflix Prize *(AAIM '08)*.

Enhance micro-blogging recommendations of posts with an homophily-based graph

Quentin Grossetti
Camelia Constantin
Cédric Du Mouza
Nicolas Travers

ABSTRACT

Due to the popularity of microblogging platforms, the amount of data procuded by users are unprecedent. One major issue is to find relevant information for each end-users, especially on real-time delivery. Faced with such a volumetry, posts with short lifetime, variety of behaviors between users and content, it becomes a real challenge for recommending systems. Traditional methods like collaborative filtering wil hardly scale up due to the high dynamicity. We present in this article a thorough study of a large Twitter dataset, focused on homophily, which leads to our recommandation approach. It relies on the construction of a similarity graph based on retweet behaviors on top of the Twitter graph. Finally we conduct experiments on our real dataset to demonstrate the quality and scalability of our method.

Maximisation en ligne et à grande échelle de l'influence sur les réseaux sociaux

Paul Lagrée
Olivier Cappe
Bogdan Cautis
Silviu Maniu

ABSTRACT

La maximisation de l'influence (IM) est le problème de la recherche d'utilisateurs / nœuds influents dans un graphe afin de maximiser la propagation de l'information. De nombreuses applications existent dans la publicité et le marketing en ligne, notamment sur les réseaux sociaux. Dans cet article, nous étudions une version générique de l'IM où il est question d'optimiser les campagnes d'influence en sélectionnant séquentiellement des nœuds d'un ensemble de candidats correspondant à un petit sous-ensemble de la population totale, sous l'hypothèse que, pour une campagne donnée, les nœuds précédemment activés restent actifs tout au long de la campagne, et n'apportent donc pas de récompenses supplémentaires. Fait important, nous ne faisons pas d'hypothèse sur le modèle de diffusion sous-jacent et nous travaillons sans connaître le réseau de diffusion ni l'historique des activations passées. Nous appelons ce problème "maximisation de l'influence en ligne avec persistance". Nous abordons d'abord les scénarios qui motivent le problème et présentons notre approche pour le résoudre. Nous introduisons un estimateur sur la masse manquante des candidats – l'espérance du nombre de nœuds qui peuvent encore être atteints à partir d'un candidat – et justifions son choix pour estimer rapidement la valeur souhaitée en s'appuyant sur des données réelles issues de Twitter. Nous décrivons un nouvel algorithme, GT-UCB, qui s'appuie sur des bornes de confiance sur la masse manquante. Nous montrons que notre approche conduit à des diffusions de bonne qualité sur des données simulées et réelles, en dépit du fait que nous ne faisons quasiment aucune hypothèse sur le support de diffusion. Enfin, notre méthode est plusieurs ordres de grandeur plus rapide que les méthodes d'IM en ligne concurrentes, ce qui nous permet d'attaquer des problèmes de grande taille auparavant inaccessibles.

Conception de Schémas de Bases de Données Relationnelles en Présence de Données Incertaines

Sebastian Link
The University of Auckland
New Zealand
s.link@auckland.ac.nz

Henri Prade
IRIT, CNRS et Université de Toulouse III
France
prade@irit.fr

ABSTRACT

On étudie l'impact de l'incertitude sur la conception de schémas de bases de données relationnelles. L'incertitude est modélisée qualitativement en assignant à chaque n-uplet le degré de possibilité avec lequel il peut être considéré comme valide, et en assignant aux dépendances fonctionnelles un degré de certitude qui réfère à quels n-uplets elles s'appliquent. Une théorie de la conception de schémas de bases de données est proposée pour les dépendances fonctionnelles possibilistes, qui inclut des caractérisations axiomatique et algorithmique efficaces pour l'inférence des dépendances. De manière naturelle, les degrés de possibilité des n-uplets conduisent à une échelle de différents degrés de redondance des données. Des versions syntactiques pondérées des Formes Normales classiques (Boyce-Codd et Troisième Forme) sont introduites et sémantiquement justifiées en termes de limitation, à différents degrés, de la redondance des données. Les techniques classiques de décomposition et de synthèse sont pondérées de même. Par suite, les dépendances fonctionnelles possibilistes ne permettent pas seulement aux concepteurs de contrôler les niveaux recherchés pour l'intégrité des données et l'absence de pertes d'information, mais aussi d'équilibrer le compromis classique entre l'efficacité de l'interrogation et celle de la mise à jour. Des expérimentations poussées confirment l'effectivité du cadre proposé et offre une approche originale à la conception des schémas relationnels.

CCS CONCEPTS

• **Information systems** → **Database design and models**; **Uncertainty**; • **Theory of computation** → *Database constraints theory*;

KEYWORDS

axiomes d'Armstrong; forme normale de Boyce-Codd; redondance de données ; troisième forme normale; théorie des possibilités

Les bases de données relationnelles ont été développées pour des applications sur des données certaines, telles qu'en comptabilité, en gestion de stocks, ou de salaires. Les applications modernes telles que l'extraction d'information, l'intégration et le nettoyage de données, requièrent des techniques autorisant des données incertaines. Les recherches sur le traitement des données incertaines ont été nombreuses, et deux directions ont été considérées: L'évaluation de requêtes a été le principal sujet d'études, et l'incertitude est alors le plus souvent modélisée quantitativement de manière probabiliste. Au contraire, l'impact de l'incertitude en conception de schémas de bases de données est appréhendé qualitativement, dans le but d'un traitement efficace des requêtes fréquentes et des mises à jour.

En conception classique de schémas, l'inefficacité dans la mise à jour est évitée en supprimant les redondances de données causée par les dépendances fonctionnnelles (DFs). Intuitivement, si une donnée est incertaine, alors il en va de même de toute redondance qui résulte de cette donnée. Plus il est possible que des redondances de données apparaissent, plus des DFs peuvent causer cette redondance, et plus l'effort de normalisation pour éliminer cette redondance est grand. En d'autres termes, l'élimination de la redondance pour des données moins plausibles (possibles) requière une normalisation par rapport à un nombre plus petit de DFs. Ceci constitue une bonne nouvelle dans la mesure où les données qui sont les moins possibles sont intuitivement les plus sujettes à des mises à jour. En conséquece, la plupart des mises à jour fréquentes peuvent être prises en charge efficacement avec un moindre effort de normalisation. Du coup, moins de normalisation conduit à une meilleure efficacité en termes d'évaluation de requêtes. Notre approche offre un cadre pratique et précis qui permet aux concepteurs de bases de données de tirer complètement avantage de cet impact intuitif de l'incertitude.

Nous modélisons l'incertitude en associant à chaque n-uplet ("tuple") un degré de possibilité (p-degré) avec lequel il est susceptible d'être présent. Le degré de possibilité appartient à une échelle finie donnée de valeurs linéairement ordonnées. Ce cadre intuitif correspond bien à la capacité des gens à raisonner qualitativement, plutôt que quantitativement. De fait, les humains aiment à classer les choses en un petit nombre de catégories et ont des difficultés à fournir des valeurs exactes telles que des probabilités. Ceci conduit à un cadre constitué par une chaîne emboitée de mondes possibles, chacun d'eux correspondant à une relation classique. Le plus petit monde ne contient que les n-uplets qui sont complètement possibles tandis que le monde le plus grand contient tous les n-uplets, à l'exclusion de ceux qui sont totalement impossibles et donc ne peuvent être présents. De plus, seul le plus petit monde est considéré comme la partie de la base de données qui soit certaine. La certitude avec laquelle une DF tient est obtenue à partir du p-degré du monde le plus petit où la DF est violée [5]. Aux extrêmes, les DFs qui sont complètement certaines tiennent même dans le monde le plus grand, et les DFs qui ne sont pas certaines du tout ne tiennent même pas dans le monde le plus petit.

Cette approche a conduit aux contributions suivantes:

1. Nous formalisons l'incertitude en attribuant des p-degrés aux n-uplets d'une base de données. L'approche est clairement fondée car elle repose sur une distribution de possibilité sur des mondes possibles qui forment une chaîne linéaire de relations.

2. Nous définissons une DF possibiliste (pDF) comme une DF classique avec un degré de certitude (c-degré), obtenu à partir du p-degré du monde possible le plus petit où la DF est violée.

3. Nous établissons une théorie complète de la conception de schémas pour les pDFs, incluant l'axiomatique et les algorithmes pour leur problème d'inférence ("implication problem").

4. Nous appliquons la théorie de la conception de schémas pour établir un nouveau cadre de normalisation, incluant des versions graduées de la forme normale de Boyce-Codd (BCNF) et de la troisième forme normale (3NF), leur justification sémantique en termes de préservation de dépendances et d'élimination de la redondance dans les données, ainsi que des algorithmes de normalisation. Nos techniques offrent aux concepteurs de base de données une série ordonnée de schémas normalisés, à partir desquels des sélections, mieux informées, du schéma final peuvent être effectuées. Nous imaginons deux scénarios principaux d'utilisation :

- Une organisation visant d'abord à préserver l'intégrité des données et l'absence de perte d'information peut choisir les DFs qui s'appliquent avec le c-degré recherché. Dans ce cas, nos algorithmes calcule les formes normales qui correspondent exactement à l'objectif.
- Pour une organisation axée sur le traitement efficace des requêtes et des mises à jour, un concepteur de base de données peut utiliser nos techniques pour obtenir les formes normales qui correspondent le mieux à la demande. Dans ce cas, il est clair que les niveaux d'intégrité de données et d'absence de pertes d'information sont atteints.

Ainsi, les c-degrés offrent un mécanisme efficace pour équilibrer les compromis entre intégrité de données et absence de perte d'information, entre efficacité de traitement des requêtes et des mises à jour. Ceci est illustré par la Figure 1, montrant comment la sélection de différents c-degrés β_i conduit à différentes décompositions, par exemple ici les β_i-3FN pour $i = 1, \ldots, k$. Nos résultats peuvent justifier classiquement des schémas dé-normalisés en termes d'autorisation de redondance des données causée par des pDFs dont le c-degré est plus petit que ce qui est recherché.

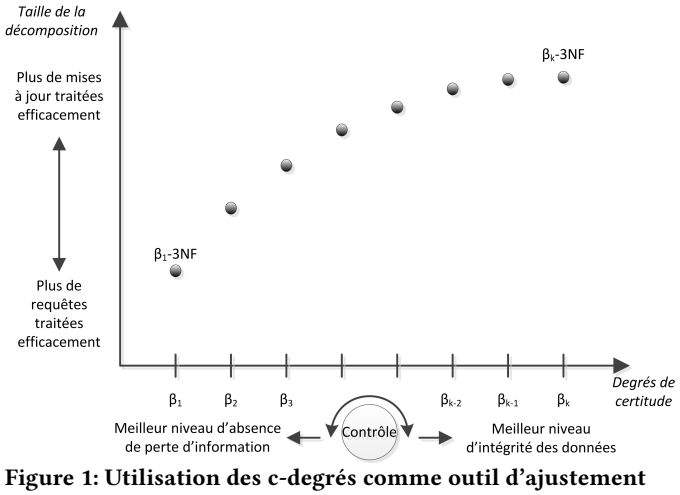

Figure 1: Utilisation des c-degrés comme outil d'ajustement

5. De nombreuses expérimentations de nos algorithmes ont confirmé leur efficacité pratique, et conduisent à un nouveau regard sur les compromis mis en œuvre en normalisation classique.

L'exemple de la table suivante et la Figure 2 associée illustrent l'idée que les niveaux de confiance dans les n-uplets induisent un ensemble ordonné de mondes possibles qui autorisent des DFs à portées variables. Pour une présentation détaillée (preuves, exemples, algorithmes et expérimentations) le lecteur pourra consulter [4, 6]. L'idée de stratification possibiliste d'une base de données a aussi été utilisée pour la gestion de clés, de contraintes de cardinalité, et le nettoyage de bases de données [1–3].

Projet	*Moment*	*Responsable*	*Salle*	*Degré poss.*
Eagle	Mon, 9am	Ann	Aqua	α_1
Hippo	Mon, 1pm	Ann	Aqua	α_1
Kiwi	Mon, 1pm	Pete	Buff	α_1
Kiwi	Tue, 2pm	Pete	Buff	α_1
Lion	Tue, 4pm	Gill	Buff	α_1
Lion	Wed, 9am	Gill	Cyan	α_1
Lion	Wed, 11am	Bob	Cyan	α_2
Lion	Wed, 11am	Jack	Cyan	α_3
Lion	Wed, 11am	Pam	Lava	α_3
Tiger	Wed, 11am	Pam	Lava	α_4

Table 1: Relation possibiliste (p-relation)

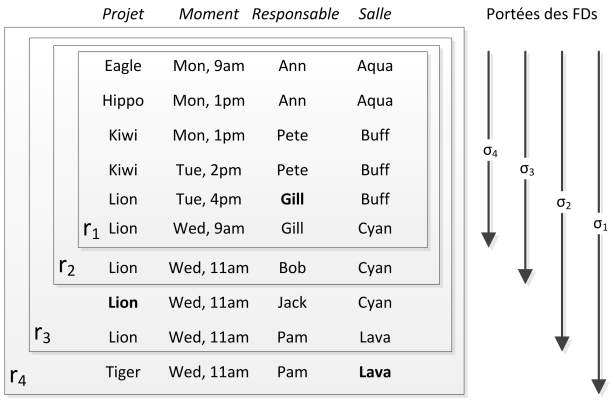

Figure 2: Mondes de la p-relation de la Table 1 avec en gras quelques valeurs redondantes et les portées des DFs

Cette recherche est soutenue par le Conseil de fonds Marsden, financé par le Gouvernement de Nouvelle Zélande, et administré par la Royal Society of New Zealand.

REFERENCES

[1] Neil Hall, Henning Köhler, Sebastian Link, Henri Prade, and Xiaofang Zhou. 2015. Cardinality constraints on qualitatively uncertain data. *Data Knowl. Eng.* 99 (2015), 126–150.

[2] Henning Köhler, Uwe Leck, Sebastian Link, and Henri Prade. 2014. Logical Foundations of Possibilistic Keys. In *JELIA*. 181–195.

[3] Henning Köhler and Sebastian Link. 2016. Qualitative Cleaning of Uncertain Data. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016.* 2269–2274.

[4] Sebastian Link and Henri Prade. 2014. *Relational database schema design for uncertain data.* Technical Report CDMTCS-469. The University of Auckland.

[5] Sebastian Link and Henri Prade. 2016. Possibilistic Functional Dependencies and Their Relationship to Possibility Theory. *IEEE Trans. Fuzzy Systems* 24, 3 (2016), 757–763.

[6] Sebastian Link and Henri Prade. 2016. Relational Database Schema Design for Uncertain Data. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016.* 1211–1220.

Efficient Scheduling of Scientific Workflows using Hot Metadata in a Multisite Cloud

Ji Liu
Microsoft Research - Inria Joint Centre, Inria, LIRMM and University of Montpellier
ji.liu@inria.fr

Luis Pineda-Morales
Microsoft Research - Inria Joint Centre, Inria and IRISA / INSA Rennes
luis.pineda-morales@inria.fr

Esther Pacitti
LIRMM, University of Montpellier and Inria
esther.pacitti@lirmm.fr

Alexandru Costan
IRISA / INSA Rennes
alexandru.costan@irisa.fr

Patrick Valduriez
Inria, LIRMM and University of Montpellier
patrick.valduriez@inria.fr

Gabriel Antoniu
Inria
gabriel.antoniu@inria.fr

Marta Mattoso
COPPE / UFRJ
marta@cos.ufrj.br

ABSTRACT

Large-scale scientific applications are often expressed as scientific workflows (SWfs). Several SWfs have huge storage and computation requirements, and so they need to be processed in multiple (cloud-federated) datacenters. It has been shown that efficient metadata handling plays a key role in the performance of computing systems. However, most of this evidence concern only single-site, HPC systems to date. In this paper, we present a hybrid distributed model and architecture, using *hot metadata* (frequently accessed metadata) for efficient SWf scheduling in a *multisite* cloud. We couple our model with a scientific workflow management system (SWfMS) to validate and tune its applicability to different real-life SWfs with different scheduling algorithms. We show that the combination of efficient management of hot metadata and scheduling algorithms improves the performance of SWfMS, reducing the execution time of highly parallel jobs up to 64.1% and that of the whole SWfs up to 37.5%.

KEYWORDS

hot metadata, metadata management, multisite clouds, scientific workflows, geo-distributed applications.

1 INTRODUCTION

Many large-scale scientific applications process amounts of data reaching the order of Petabytes; as the size of the data increases, so do the requirements for computing resources. Clouds stand out as convenient infrastructures for handling such applications, for they offer the possibility to lease resources at a large scale and relatively low cost. Very often, requirements of data-intensive scientific applications exceed the capabilities of a single cloud datacenter (site), either because the site imposes usage limits for fairness and security [2], or simply because the dataset is too large. Also, the application data are often physically stored in different geographic locations, because they are sourced from different experiments, sensing devices or laboratories. Hence multiple sites are needed in order to guarantee both that enough resources are available and that data are processed as close to its source as possible. All popular public clouds today account for geo-distributed datacenters.

A large number of data-intensive distributed applications are expressed as Scientific Workflows (SWf). A SWf is an assembly of scientific data processing activities with data dependencies between them [4]. The application is modelled as a graph, in which vertices represent processing jobs, and edges their dependencies. A job may correspond to multiple tasks during execution.

Currently, many Scientific Workflow Management Systems (SWfMS) are publicly available [5, 16]; some of them already support multisite execution [10–12]. In order to enable SWf execution in a multisite cloud with distributed input data within a reasonable time, the execution of the tasks of each job should be efficiently scheduled to a corresponding site. Thus, the multisite scheduling process is to use scheduling algorithms to decide at which site to execute the tasks to achieve a given objective, *e.g.* reducing execution time.

Most of the existing SWf engines focus on the optimizations brought to the *data* management layer. In multisite environments, such solutions privilege starting the SWf only after gathering all the needed data in a shared-disk file system at one data center, which is time consuming. Less attention was given to the *metadata*. In most data processing systems, metadata are typically stored, managed and queried at some centralized servers located at a centralized site [5, 7, 14], *i.e.* a centralized strategy. However, this strategy will be inefficient if the number of queries is high or the bandwidth is low. In addition, few attention was paid to the combination of metadata management strategies and scheduling algorithms.

In this paper, we take a different approach based on the distribution of hot metadata (frequently accessed metadata) in order to increase the locality of access by different computing nodes. First, we distinguish hot metadata. Then, we adapt two distributed metadata management strategies, *i.e.* Distributed Hash Table (DHT) and

replication method (REP), for hot metadata management, and a local storage based hot metadata management strategy, *i.e.* LOC, to manage the hot metadata. Finally, we combine the strategies with scheduling algorithms.

This paper is organized as follows. Section 2 introduces the hot metadata, presents the design principles, proposes hot metadata management strategies. Section 3 presents our experimental results. Finally, section 4 concludes.

2 OUR APPROACH: DISTRIBUTED HOT METADATA

Metadata management significantly impacts the performance of computing systems dealing with thousands or millions of individual files. In this section, we first introduce hot metadata. Then, we discuss design principles for our architecture, including the combination of metadata management strategies and scheduling algorithms. Afterwards, we present the distributed metadata management strategies.

Metadata have a critical impact on the efficiency of SWf execution [3, 15]. They should be readily available to the system at any given time to provide a global view of data location and to enable task tracking during the execution. Most notably, we assert that some metadata are more frequently accessed than others. We denote such metadata by *hot metadata* and argue that it should be handled in a specific, more quickly accessible way than the rest of the metadata. We empirically distinguish two types of hot metadata: *task metadata* and *file metadata*.

Several key choices set up the foundation of our architecture. First, we propose to use a two-layer multisite system: The lower *intra-site* layer operates as current single-site SWfMS and a higher *inter-site* layer coordinates the interactions at site-level through a master/slave architecture. Second, as maintaining an updated version of all metadata across a multisite environment consumes a significant amount of time, we evaluate different storage strategies for hot metadata, while keeping cold metadata stored locally and synchronizing cold metadata only during the execution of the job. Third, our system do not guaranty a fully consistent state while the system is guaranteed to be eventually consistent. Finnaly, as the scheduling process may need to get the information about where the input data of each task is located, which is available in the hot metadata, we directly cache the file metadata at the coordinator site for scheduling [9].

We consider two different alternatives, *i.e.* DHT and REP [13], for decentralized metadata management and propose a local storage based hot metadata management strategy, *i.e.* LOC. LOC stores the hot metadata at the site where it is generated. DHT stores the hot metadata at the site corresponding to its hash value and REP stores the hot metadata to the sites where it is generated and to the site corresponding to its hash value.

3 EXPERIMENTAL EVALUATION

We use Multisite Chiron [10] deployed on the Microsoft Azure cloud [1] with a total of 27 nodes of A4 standard virtual machines (VMs) to validate our approach. The VMs were evenly distributed among three datacenters. We take two real-life SWfs, *i.e.* Montage [8] and Buzz [6] as use case. The experimental results reveal that LOC srategy can be up to 28% better than the centralized method in terms of overall multisite SWf execution and the combination of LOC and DIM can reduce the overall SWf execution time up to 37.5%. In addition, the gain of the execution time of a multi-task job can be up to 64.1% for the REP strategy.

4 CONCLUSION

In this paper, we introduced the concept of hot metadata for SWfs running in large, geographically distributed and highly dynamic environments. Based on it, we designed a hybrid decentralized and distributed approach for handling metadata in multisite clouds. The approach includes a distributed architecture and two adapted and one proposed hot metadata management strategies. We validated the approach by executing real life SWfs in a multisite cloud. Our experimental results showed an improvement of up to 37.5% for the whole SWf's execution time and 64.1% for specific highly-parallel jobs, compared to state-of-the-art centralized solutions. Encouraged by these results, we plan to broaden the scope of our work and consider the impact of heterogeneous multisite environments and integrating real-time monitoring information.

REFERENCES

[1] Microsoft Azure Cloud. http://www.windowsazure.com/en-us/.

[2] Resource Quotas - Google Cloud Platform. https://cloud.google.com/compute/docs/resource-quotas.

[3] Sadaf R. Alam, Hussein N. El-Harake, Kristopher Howard, Neil Stringfellow, and Fabio Verzelloni. Parallel I/O and the metadata wall. In *Proc. of the 6th Workshop on Parallel Data Storage*, PDSW '11, pages 13–18, New York, NY, USA, 2011. ACM.

[4] E. Deelman, D. Gannon, M. Shields, and I. Taylor. Workflows and e-science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528–540, 2009.

[5] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, 13(3):219–237, 2005.

[6] J. Dias, E. Ogasawara, D. De Oliveira, F. Porto, P. Valduriez, and M. Mattoso. Algebraic dataflows for big data analysis. In *Big Data, 2013 IEEE Intl. Conf. on*, pages 150–155, 2013.

[7] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5):29–43, October 2003.

[8] Gideon Juve, Ann Chervenak, Ewa Deelman, Shishir Bharathi, Gaurang Mehta, and Karan Vahi. Characterizing and profiling scientific workflows. *FGCS*, 29(3):682 – 692, 2013.

[9] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso. Scientific workflow scheduling with provenance data in a multisite cloud. *Transactions on Large-Scale Data- and Knowledge-Centered Systems*, 2016. to appear.

[10] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso. Scientific workflow scheduling with provenance support in multisite cloud. In *High Performance Computing for Computational Science VECPAR*, 2016.

[11] J. Liu, E. Pacitti, P. Valduriez, D. De Oliveira, and M. Mattoso. Multi-objective scheduling of scientific workflows in multisite clouds. *Future Generation Computer Systems*, 63:76–95, 2016.

[12] J. Liu, V. S. Sousa, E. Pacitti, P. Valduriez, and M. Mattoso. Scientific workflow partitioning in multisite cloud. In *Euro-Par 2014: Parallel Processing Workshops - Euro-Par 2014 Int. Workshops*, pages 105–116, 2014.

[13] L. Pineda-Morales, A. Costan, and G. Antoniu. Towards multi-site metadata management for geographically distributed cloud workflows. In *IEEE Intl. Conf. on Cluster Computing*, pages 294–303, Sept 2015.

[14] Frank Schmuck and Roger Haskin. GPFS: A shared-disk file system for large computing clusters. In *Proc. of the 1st USENIX Conference on File and Storage Technologies*, FAST '02, Berkeley, CA, USA, 2002.

[15] Alexander Thomson and Daniel J Abadi. CalvinFS: consistent wan replication and scalable metadata management for distributed file systems. In *Proc. of the 13th USENIX Conf. on File and Storage Technologies*, 2015.

[16] J. M. Wozniak, T. G. Armstrong, M. Wilde, D. S. Katz, E. L. Lusk, and I. T. Foster. Swift/t: scalable data flow programming for many-task applications. In *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 309–310, 2013.

Apprentissage automatique de règles CEP prédictives: combler le gap entre fouille de données et traitement des événements complexes

Raef Mousheimish
Yehia Taher
Karine Zeitouni

ABSTRACT

En raison de l'avantage indéniable de la prédiction et de la proactivité, de nombreux domaines de recherche et des applications industrielles s'orientent de plus en plus vers les sciences de données et en particulier l'analyse prédictive. Cependant, en raison de trois faits bien connus, l'aspect réactif du traitement des événements complexes (CEP) va à l'encontre de l'exigence de la prédiction. Le premier fait est que le mécanisme d'inférence dans ce domaine est entièrement guidé par les règles CEP. Le second est que la seule façon de définir une règle CEP est qu'un expert puisse la formuler, ce qui n'est pas toujours simple à réaliser. Le troisième fait est que les experts ont tendance à définir des règles CEP réactives, et ce en raison de la difficulté à exprimer des règles CEP pédictives. En combinant ces faits, le CEP a été jusque-là limité au contexte réactif à des évènements observés. Dans cet article, nous présentons une nouvelle approche basée sur la fouille de données qui apprend automatiquement des règles CEP prédictives guidée par les données historiques. L'approche propose un nouvel algorithme d'apprentissage où des modèles complexes à partir de séries temporelles multivariées sont appris. Ensuite, au moment de l'exécution, une transformation automatique permet l'expression de règles CEP prédictive et l'instancihtion d'un moteur CEP prêt à être exécuté. De nombreuses expérimentations ont été réalisée sur des ensembles de données publiques démontrant l'efficacité de notre approche.

Complexity of Certain Query Answering on Hyperstreams

Complexité du calcul des réponses certaines sur les hyperflux

Iovka Boneva
Université de Lille
Links team
iovka.boneva@univ-lille.fr

Joachim Niehren
Inria, Lille
Links team
joachim.niehren@inria.fr

Momar Sakho
Inria, Lille
Links team
momar.sakho@inria.fr

ABSTRACT

This article is an excerpt of the longer paper [1], published in the RP'18 conference and which can be found at https://hal.inria.fr/hal-01846016.

A hyperstream is a sequence of streams with references to others. We study the complexity of computing certain answers for queries evaluated on hyperstreams that describe strings. We show that the problem is PSPACE-complete for queries defined by deterministic finite automata, but that it can be solved in P for linear hyperstreams even in the presence of compression.

1 INTRODUCTION

A stream is a sequence of events that arrive incrementally one by one from the left to the right. Most typically, streams are produced by social networks such as Twitter, database systems as for producing financial transactions, information systems, sensor systems, or more generally when communicating semi-structured data over the internet. We are interested in the problem of monitoring streams in a reactive manner [2–5]. The objective is to select the relevant events of a stream as quickly as possible upon their arrival. This requires to decide whether an event of the stream is a certain answer of the logical query that defines the relevant events of the monitoring task. Lowering the latency of this decision process increases the reactivity of the stream processing system and reduces its memory costs. A limitation to constant memory may seem ideal in theory, but is too restrictive for many monitoring tasks in practice. A less restrictive objective is thus to minimize the latency and thereby to reduce the memory consumption.

In the present paper we study a generalization of streams to multiple streams with references as introduced by Maneth, Ordóñez and Seidl [6]. The references point to unknown parts in the middle of a stream. The same reference may be used multiple times, allowing to share unknown parts. Streams with similar references were named hyperstreams in own previous work [7]. Here, we propose to formalize hyperstreams containing words (rather than linearizations of trees or nested words) as *incomplete* versions of singleton context-free grammars [8] (also termed straight line programs [9]), where the rules of some nonterminals may be missing. The hyperstream $G_1 = (\Sigma, N, R, S)$ is illustrated graphically in Fig. 1. It has the terminals in $\Sigma = \{a, b, c\}$, the nonterminals in $N = \{S, X, Y, Z\}$, the set R with the rules $S \to aXbbYaX$ and $X \to YcZa$, and the start symbol S. The nonterminals are called the references of the hyperstream. For some of these references there exists a rule in the grammar, and if so, this rule is unique. For any grammar rule, the reference on its left is said to refer to the string pattern on its right. The hyperstream G_1 has a rule for S and X, while it misses those of Y and Z. The missing rules for these references may be added in the future one by one by the hyperstream's environment. Alternatively, hyperstreams can be identified with *compressed string patterns*. The hyperstream G_1 for instance represents the string pattern $p_1 = a\underline{YcZa}bbYa\underline{YcZa}$, while sharing the underlined factors substituted for the two occurrences of X. Streams are a special case of string patterns that have a unique occurrence of a variable in their last position. The landscape from streams to hyperstreams, over linear string patterns, string patterns, and compressed string patterns is illustrated in Fig. 2.

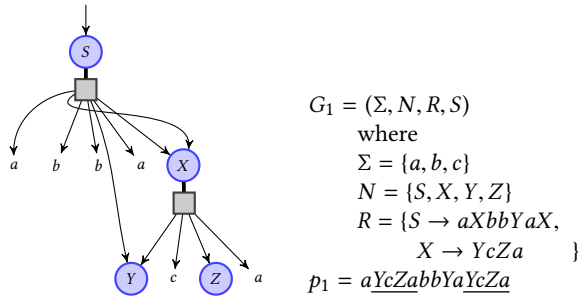

Figure 1: The hyperstream G_1 and its string pattern p_1.

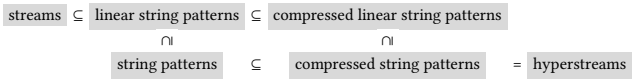

Figure 2: Landscape from streams to hyperstreams.

In this paper, we study the decision problem of certain query answering (CQA) on compressed string patterns, i.e., whether a tuple of positions is a certain answer of a query on a compressed string pattern. Here we consider the positions of the string pattern after decompression rather than the positions of the grammar. Intuitively, a tuple of positions is a certain query answer on a compressed string pattern G if it is an answer to the query on all completions of G, up to the offsets raised by the completion of G on its decompression. We will also consider the symmetric problem for certain query non-answers.

Motivated by regular path queries [10], we consider nondeterministic finite automata (NFAs) for defining such queries. For instance, the query $\mathbf{Q}_1$ on strings over $\Sigma = \{a, b, c\}$ that selects all

a-positions that are followed eventually by bb can be defined by the following regular path in XPath-like notation:

$$\mathbf{Q}_1 = \texttt{successor}^*::a[\texttt{successor}^*::b/\texttt{successor}::b].$$

It can also be defined by the x-pointed regular expression $\Sigma^* a^x \Sigma^* bb\Sigma^*$ where x is a variable for the position that is to be selected. Now, consider the case, where the string is not given explicitly but only described partially by some (compressed) string pattern. On the string pattern p_1, for instance, the a-positions 1 and 5 are certain query answers for $\mathbf{Q}_1$, while the a-positions 9 and 13 are not. The position 13 is even a certain non-answer.

When restricted to Boolean Nfa queries, Cqa becomes equivalent to the problem of whether all strings described by the completions of a compressed string pattern are accepted by the Nfa. For the string pattern Y (for some variable Y), this problem clearly generalizes on the universality problem of Nfas, which is well-known to be Pspace-complete [11]. The following questions, however, are open to the best of our knowledge, even in the case of string patterns without compression: Is Cqa on (compressed) string patterns decidable for Nfa-defined queries, and if yes, what is the complexity? Does Cqa on (compressed) string patterns remain hard for queries defined by deterministic finite automata (Dfas)? For which restrictions of (compressed) string patterns is Cqa in P? And what about the symmetric questions concerning certain query non-answers? The objective of the present paper is to answer these questions in all possible cases.

Our first contribution is that Cqa on string patterns is Pspace-complete, both for Nfa queries and Dfa queries, with and without compression, Boolean or not, see Fig. 3. This upper bound is not fully obvious, as the set of strings defined by a string pattern may be non-regular and even non-context-free. Furthermore, the lower bound may be surprising in that Cqa for Dfa queries on string patterns is more complex than on streams, where it is in P (Theorem 1 of [12]), and also more complex than string pattern matching, which is NP-complete (Theorem 3.6 of [13]) even with compression (Theorem 4.10 of [14]).

Our second contribution is that Cqa for Dfa queries can be decided in P on compressed *linear* string patterns, see Fig. 4. The linearity restriction matches with the worst case complexity for streams, even though linear compressed string patterns allow for unknown factors and compression in addition. This result is based on a novel algorithm for partial decompression of compressed string patterns that we present followed by a test of a reachability property.

Our third contribution is that the certainty of query non-answers on compressed string patterns is Pspace-complete, both for Boolean and non-Boolean queries, and independently of whether they are defined by Dfas or Nfas. In the Boolean case, the problem is equivalent to whether a compressed string pattern does *not* match the regular language accepted by the automaton. This problem generalizes on the complement of compressed string pattern matching, and thus is coNP-hard. So while certain query non-answering can be solved in P on streams, the complexity increases to Pspace on compressed string patterns. Finally, we show that the restriction of the problem to compressed linear string patterns – that is, *regular* compressed linear string pattern matching – can also be solved in P even for queries defined by Nfas.

	Dfas	Nfas
Answers	Pspace-c	Pspace-c
Non-answers	Pspace-c	Pspace-c

Figure 3: Query certainty on (compressed) string patterns.

	Dfas	Nfas
Answers	P	Pspace-c
Non-answers	P	P

Figure 4: Query certainty on (compressed) linear string patterns.

REFERENCES

[1] Iovka Boneva, Joachim Niehren, and Momar Sakho. Certain query answering on compressed string patterns: From streams to hyperstreams. In Igor Potapov and Pierre-Alain Reynier, editors, *Reachability Problems - 12th International Conference, RP 2018, Marseille, France, September 24-26, 2018, Proceedings*, volume 11123 of *Lecture Notes in Computer Science*, pages 117–132. Springer, 2018.

[2] Barzan Mozafari, Kai Zeng, and Carlo Zaniolo. High-performance complex event processing over XML streams. In K. Selçuk Candan, Yi Chen, Richard T. Snodgrass, Luis Gravano, Ariel Fuxman, K. Selçuk Candan, Yi Chen, Richard T. Snodgrass, Luis Gravano, and Ariel Fuxman, editors, *SIGMOD Conference*, pages 253–264. ACM, 2012.

[3] Michael Kay. A streaming XSLT processor. In *Balisage: The Markup Conference 2010. Balisage Series on Markup Technologies*, volume 5, 2010.

[4] Michael Schmidt, Stefanie Scherzinger, and Christoph Koch. Combined static and dynamic analysis for effective buffer minimization in streaming XQuery evaluation. In *23rd IEEE International Conference on Data Engineering*, pages 236–245, 2007.

[5] Dan Olteanu. SPEX: Streamed and progressive evaluation of XPath. *IEEE Trans. on Know. Data Eng.*, 19(7):934–949, 2007.

[6] Sebastian Maneth, Alberto Ordóñez Pereira, and Helmut Seidl. Transforming XML streams with references. In Costas S. Iliopoulos, Simon J. Puglisi, and Emine Yilmaz, editors, *String Processing and Information Retrieval - 22nd International Symposium, SPIRE 2015, London, UK, September 1-4, 2015, Proceedings*, volume 9309 of *Lecture Notes in Computer Science*, pages 33–45. Springer, 2015.

[7] Pavel Labath and Joachim Niehren. A functional language for hyperstreaming XSLT. Technical report, INRIA Lille, 2013.

[8] Wojciech Plandowski. *The complexity of the morphism equivalence problem for context-free languages*. PhD thesis, Warsaw University. Department of Informatics, Mathematics, and Mechanics, 1995.

[9] L. Babai and E. Szemeredi. On the complexity of matrix group problems i. In *Proceedings of the 25th Annual Symposium onFoundations of Computer Science, 1984*, SFCS '84, pages 229–240, Washington, DC, USA, 1984. IEEE Computer Society.

[10] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan L. Reutter, and Domagoj Vrgoc. Foundations of modern graph query languages. *CoRR*, abs/1610.06264, 2016.

[11] Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 254–266. IEEE Computer Society, 1977.

[12] Olivier Gauwin, Joachim Niehren, and Sophie Tison. Earliest Query Answering for Deterministic Nested Word Automata. In *17th International Symposium on Fundamentals of Computer Theory*, volume 5699 of *LNCS*, pages 121–132, Wraclaw, Poland, September 2009. Springer Verlag.

[13] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.

[14] Adria Gascón, Guillem Godoy, and Manfred Schmidt-Schauß. Context matching for compressed terms. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science, LICS 2008, 24-27 June 2008, Pittsburgh, PA, USA*, pages 93–102. IEEE Computer Society, 2008.

Énumération des requêtes du premier ordre sur classes de bases de données avec local bounded expansion

Luc Segoufin
Alexandre Vigny

ABSTRACT

We consider the evaluation of first-order queries over classes of databases with *local bounded expansion*. This class was introduced by Nešetřil and Ossona de Mendez and generalizes many well known classes of databases, such as bounded degree, bounded tree width or bounded expansion. It is known that over classes of databases with local bounded expansion, first-order sentences can be evaluated in pseudo-linear time (pseudo-linear time means that for all ϵ there exists an algorithm working in time $O(n^{1+\epsilon})$). Here, we investigate other scenarios, where queries are not sentences. We show that first-order queries can be enumerated with constant delay after a pseudo-linear preprocessing over any class of databases having locally bounded expansion. We also show that, in this context, counting the number of solutions can be done in pseudo-linear time.

Partage de documents sécurisé dans le Cloud Personnel

Paul Tran-Van 1
[1] Cozy Cloud, France
paul@cozycloud.cc

Nicolas Anciaux 2,3
[2] Inria, France
nicolas.anciaux@inria.fr

Philippe Pucheral 2,3
[3] U. Versailles St-Q., France
philippe.pucheral@uvsq.fr

1. INTRODUCTION

La numérisation croissante de nos vies a favorisé l'émergence d'un marché focalisé sur l'analyse des données personnelles afin d'établir des profils de plus en plus poussés et intrusifs. Parallèlement, des surveillances d'états se mettent en place qui font craindre un glissement progressif vers des dystopies jusqu'ici réservées à la littérature. Afin de répondre à cette situation, le paradigme du Cloud Personnel (CP) s'est développé : chaque utilisateur a désormais la possibilité de stocker et gérer l'intégralité de son patrimoine numérique dans un unique espace de confiance sous son contrôle. Cela entraîne cependant un changement de gouvernance sur les données, dont la sécurité et l'administration reposent désormais sur les épaules des individus. En particulier lorsqu'ils souhaitent partager leurs documents et donc les exposer à des personnes ou services tiers. Nous proposons ainsi un nouveau paradigme dans la façon de partager dans le CP au travers de deux contributions : (i) une architecture *Privacy-by-Design* et (ii) un modèle de partage adapté aux propriétés du CP.

2. ARCHITECTURE DE REFERENCE

De nombreuses approches ont été envisagées autour du partage de données et de son application sécurisée [2, 3]. Cependant, aucune n'a été adoptée à grande échelle et toutes échouent à adresser les spécificités du CP.

La première difficulté vient de la complexité du moniteur de référence en charge du contrôle d'accès : l'exécution à la volée de requêtes qui peuvent porter sur des attributs ou des rôles [1] permet une certaine flexibilité dans la définition du contrôle d'accès, mais complexifie l'expression générale de la politique de partage et ses potentiels effets. Le propriétaire doit avoir une entière confiance à la fois dans le code exécuté et dans la plateforme qui le fait tourner. Or ces deux vecteurs de confiance ne sont pas toujours justifiés : d'une part, plus un système est complexe et plus les probabilités d'erreurs sont élevés ; et d'autre part, aucune garantie tangible n'est fournie vis-à-vis de l'environnement d'exécution du moniteur de référence, ce qui le rend potentiellement vulnérable.

La seconde difficulté est liée au fait que les propriétaires se retrouvent démunis pour correctement administrer leurs partages, afin de vérifier quelles sont exactement les permissions accordées et s'assurer que leur modèle mental en terme de dissémination des données correspond bien à ce qui est réellement appliqué, ce qui est en réalité rarement le cas. La plupart du temps, les utilisateurs partagent plus que ce dont ils ont conscience [2]. Le risque est donc grand de voir les propriétaires déléguer l'administration de leurs données à des fournisseurs de services centralisés pour se décharger de cette responsabilité. Cela brise le principe d'*empowerment* du CP mais surtout aggrave le phénomène de centralisation des données, en fournissant à l'hébergeur potentiellement l'intégralité du patrimoine numérique.

Pour résoudre cette situation contradictoire, nous proposons une architecture *Privacy-by-Design* pour le CP qui peut être déclinée en de multiples instances sécurisées. Des outils d'administration sont également proposés afin que le propriétaire puisse avoir conscience des effets concrets de sa politique de partage et facilement contrôler qu'elle est correctement appliquée sans avoir besoin de fournir un effort cognitif démesuré.

Cette architecture, présentée en Figure 1, distingue trois composantes principales, chacune ayant différentes hypothèses en termes de confiance et de sécurité :

Untrusted Environment (UE) sur laquelle aucune garantie de sécurité n'existe. C'est ici que la plateforme de CP est exécutée et les données y sont stockées chiffrées.

Isolated environment (IE) où du code arbitraire peut être exécuté sans garantie sur sa fiabilité ou son honnêteté, mais avec la garantie qu'aucune information ne peut fuiter. On y exécute les règles de partages (qui peuvent provenir de sources tierces) dans le *Policy Translator* ainsi que les interfaces d'administration (*Viewers* et *Admin. GUI*).

Secure Execution Environment (SEE) qui exécute uniquement des programmes certifiés et protège les données et le code contre l'espionnage et les attaques physiques. Le moniteur de référence y est exécuté, représenté par la fonction *Allowed (sujet, document, action)* qui évalue si une *Access Control List (ACL)* existe pour chaque demande d'accès.

Chaque ACL peut être dans quatre états distincts dans le SEE : ACL^{*}, $ACL^{?}$, ACL^{-} ou ACL^{+}. Les ACL^{*} représentent les ACL non encore évaluées par la console d'administration. Cette dernière détermine si une ACL générée par une règle de partage est suspecte ($ACL^{?}$), refusée (ACL^{-}) ou acceptée (ACL^{+}), ces dernières étant les seules prises en compte par la fonction *Allowed*. En effet, une règle de partage peut être potentiellement complexe et produire des ACL qui peuvent ne pas correspondre à la politique de partage souhaitée, ou même être malveillante. La console d'administration a donc pour rôle de placer en quarantaine toute ACL suspectes et ainsi se prémunir contre des fuites de données.

Plusieurs méthodes sont envisagées pour déterminer si une ACL est suspecte, mais ne sont pas détaillées ici. On peut par exemple utiliser la sensibilité d'un document ou sujet (via les watchdog triggers) ou se fonder sur les habitudes de partage [7].

Une fois détectées, les ACL suspectes doivent ensuite être manuellement classifiées en ACL^{-} ou ACL^{+} par le propriétaire du CP via les interfaces d'administration : cela implique que tout sujet et document doit pouvoir être visualisable.

Cette architecture est plus grandement détaillées dans [7] et a fait l'objet d'un démonstrateur utilisant un hardware sécurisé comme *SEE* [5].

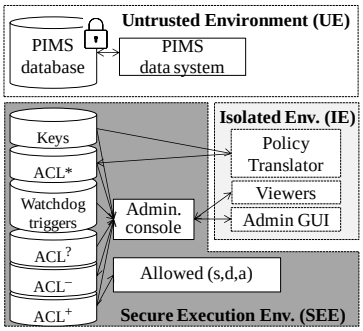

Fig. 1: Architecture

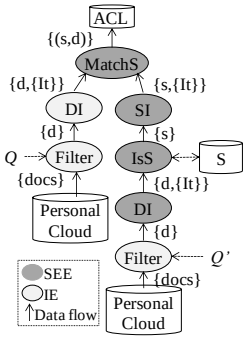

Fig. 2: Production d'ACL

3. MODELE DE PARTAGE

L'adoption d'un nouveau paradigme passe inéluctablement par de nouveaux usages innovants et simples d'utilisation. Or, les spécificités du Cloud personnel nécessitent la définition d'un nouveau modèle de partage, capable de tirer parti de la richesse et de la diversité des données présentes. Nous discutons dans cette section de la partie modèle, complémentaire à l'architecture.

Dans le CP, des permissions peuvent être induites directement du contenu des documents à partager. Dit autrement, certains documents font apparaître naturellement des sujets qui sont implicitement légitimes pour être des cibles d'une action de partage. Ceci nous amène au premier principe fondamental de notre modèle : **les documents sont des règles.**

Afin de pouvoir exprimer ces permissions, les sujets directement concernés par ces documents doivent pouvoir être considérés dans la définition de la règle grâce à des traits d'identification (*IT*). Nous appelons règles réflexives les règles qui se fondent sur ce principe. Un corollaire est que chaque sujet doit correspondre à un document du Cloud personnel, par exemple, une fiche contact, un CV, etc. Cela donne corps à un autre principe clé du modèle : **les sujets sont des documents.**

De même, ce qui est partagé doit toujours correspondre à des documents visualisables, comme cela a été dit précédemment. Même si l'objet du partage est le résultat d'un calcul complexe sur un ensemble de documents, par exemple une série temporelle de consommation électrique, ce résultat doit pouvoir s'exprimer sous la forme d'un document intelligible pour le propriétaire. Cela correspond ici au dernier principe du modèle : **les objets sont des documents**.

Ainsi, SWYSWYK se fonde sur la matérialisation de toutes les ACL, ce qui garantit une évaluation triviale de la fonction *Allowed* et permet au propriétaire de visualiser et filtrer les effets de ses règles de partage. Nous détaillons ici comment les règles sont créées et maintenues.

Cinq opérateurs sont nécessaires pour exprimer n'importe quelle règle réflexive, à savoir *Filter, DI, SI, IsS et MatchS* qui s'exécutent de façon ensembliste. Par exemple, l'opérateur *Filter* s'applique à tous les documents du CP et retourne le sous-ensemble des documents satisfaisant la condition Q. La séquence de données consommées et produite par les opérateurs est présentée en Figure 2, pour la traduction de règles réflexives en ACL.

Lors de la création d'une règle, le *Policy translator* doit l'appliquer sur tous les documents du CP. Un opérateur *Filter* est appliqué aux feuilles de chaque branche de l'arbre d'évaluation, ce qui correspond à une sélection de sujets sur la branche de droite et une sélection de documents sur la branche de gauche. Puis, l'opérateur *DI* extrait la liste des *IT* depuis les documents des sujets ciblés et les transmet à *IsS* qui tente de faire correspondre ces *IT* à ceux des sujets déjà enregistrés dans la base des sujets *S. L*'opérateur *IsS* peut avoir pour effet de bord la mise à jour dynamique de *S* quand des sujets inconnus sont rencontrés.

L'opérateur *MatchS* prend en entrée les *IT* retournés par les opérateurs *DI*, pour les documents, et *SI* pour les sujets et fait une jointure afin de produire les ACL.

A chaque fois qu'un document *d* est supprimé du Cloud personnel, toute entrée qui le référence dans *les ACL* doit l'être également. De plus, tout nouveau document *d* déclenche la réévaluation des *Filters* de chaque branche de l'ensemble des règles.

IsS, MatchS et *SI*, en gris foncé sur la Figure 2, sont des opérateurs sécurisés qui s'exécutent directement dans le *SEE* de l'architecture. A l'inverse, les *Filters* et *DI,* coloriés en gris clair*,* sont des UDF, dans lesquels la confiance est relative car pouvant provenir de sources variées. Ils sont donc exécutés en isolation par le *Policy Translator* pour prévenir toute fuite de données provoquée par un code malveillant ou mal conçu.

Le modèle de partage brièvement présenté ici est plus longuement détaillé dans [6] et a fait l'objet d'un démonstrateur intégré à la plateforme de CP Cozy [4] qui utilise une règle de partage exploitant la reconnaissance faciale sur les photos.

4. CONCLUSION

Nous espérons que ces travaux ouvriront à la voie à de nouvelles perspectives autour du partage de données respectueux de la vie privée. La vision de l'architecture a déjà été approfondie dans [1] qui identifie différents challenges à résoudre, à la fois académiques et techniques pour permettre une meilleure régulation dans la gestion des données personnelles.

Le modèle de partage à quant à lui été partiellement implémenté dans le protocole de partage open-source utilisé dans Cozy, permettant de partager des répertoires et albums photos en pair-à-pair avec ses contacts.

REFERENCES

[1] Anciaux, N., Bonnet, P., Bouganim,*et al* . (2018). Personal Data Management Systems: The security and functionality standpoint.In Information System.

[2] Bertino, E., Gabriel G., & Ashish K. "Access control for databases: Concepts and systems." Foundations and Trends® in Databases, 2011

[3] Mazurek, M. L., Arsenault, J. P., Bresee, *et al.* (2010). Access control for home data sharing: Attitudes, needs and practices. In SIGCHI (pp. 645-654).

[4] Thilakanathan, D., Chen, S., Nepal, S., & Calvo, R. A. (2014). Secure data sharing in the Cloud. In Security, PTCS (pp. 45-72).

[5] Tran-Van, P., Anciaux, N., & Pucheral, P. (2018). Reconciling Privacy and Data Sharing in a Smart and Connected Surrounding. In EDBT.

[6] Tran-Van, P., Anciaux, N., & Pucheral, P. (2017). SWYSWYK: a new Sharing Paradigm for the Personal Cloud. In ADMA (pp. 839-845).

[7] Tran-Van, P., Anciaux, N., & Pucheral, P. (2017). A new sharing paradigm for the personal cloud. In TrustBus (pp. 180-196).

[8] Tran-Van, P., Anciaux, N., & Pucheral, P. (2017). SWYSWYK: a privacy-by-design paradigm for personal information management systems. In ISD.

Massively Distributed Environments and Closed Itemset Mining the DCIM Approach

Mehdi Zitouni
Reza Akbarinia
Sadok Ben Yahia
Florent Masseglia

ABSTRACT

Data analytics in general, and data mining primitives in particular, are a major source of bottlenecks in the operation of information systems. This is mainly due to their high complexity and intensive call to IO operations, particularly in massively distributed environments. Moreover, an important application of data analytics is to discover key insights from the running traces of information system in order to improve their engineering. Mining closed frequent itemsets (CFI) is one of these data mining techniques, associated with great challenges. It allows discovering itemsets with better efficiency and result compactness. However, discovering such itemsets in massively distributed data poses a number of issues that are not addressed by traditional methods. One solution for dealing with such characteristics is to take advantage of parallel frameworks like, *e.g.*, MapReduce. We address the problem of distributed CFI mining by introducing a new parallel algorithm, called DCIM, which uses a prime number based approach. A key feature of DCIM is the deep combination of data mining properties with the principles of massive data distribution. We carried out exhaustive experiments over real world datasets to illustrate the efficiency of DCIM for large real world datasets with up to 53 million documents.

Résumés des articles courts

End-to-end Graph Mapper

Benjamin Billet
Inria, Sophia-Antipolis, France
benjamin.billet@inria.fr

Mickaël Jurret
Beepeers, Sophia-Antipolis, France
mickael.jurret@beepeers.fr

Didier Parigot
Inria, Sophia-Antipolis, France
didier.parigot@inria.fr

Patrick Valduriez
Inria, Sophia-Antipolis, France
patrick.valduriez@inria.fr

ABSTRACT

The growth of linked data in web and mobile applications motivates software developers to model their business data as graphs, enabling them to leverage the capabilities of various graph databases. Going one step further, we introduce an End-to-end Graph Mapper (EGM) for modeling the whole application as (i) a set of graphs representing the business data, the in-memory data structure maintained by the application and the user interface (tree of graphical components), and (ii) a set of standardized mapping operators that maps these graphs with each other. As a benefit, the application becomes a complex live query over multiple graph databases, making the development process simpler and safer, thanks to the automation of repetitive development tasks.
This work is done in collaboration with Beepeers (`www.beepeers.com`), a startup that develops and markets social network mobile applications for small communities.

CCS CONCEPTS

• **Information systems** → Graph-based database models; • **Software and its engineering** → Model-driven software engineering;

KEYWORDS

Graph Databases; Object-Graph Mapping; Linked Data; Software Development

1 INTRODUCTION

Nowadays, an increasing number of mobile and web applications deal with linked data (e.g., social networks, online stores, recommendation systems) which leads developers to model their data as graphs. In this context, it is natural to use a Graph Database Management System (GDBMS) as an alternative to a relational DBMS, since it provides (i) dedicated data structures for storing nodes, links and key/value pairs efficiently, and (ii) query engines for browsing these structures easily [1].

In practice, web and mobile applications are typically composed of a client part and a server part that communicate with each other using web services. In a nutshell, the client part provides a graphical interface for the user while the server part manages the application logic and communicates with a standalone database for storing the linked data. As illustrated in Figure 1, each part of the application

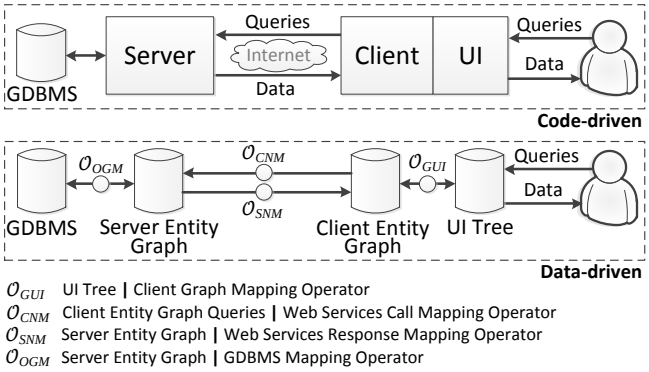

Figure 1: Representing applications as mapping operators.

relies on dedicated data structures managed by specialized data management systems: for example, the user interface manages a tree of queryable graphical components (e.g., the document object model[1], in the context of web interfaces), the client application maintains a queryable cache containing offline server data or a data store specialized for specific types of applications (e.g., the knowledge base in Yarta) [3], and the server application manages queryable in-memory entities that are mapped onto actual database entities (e.g., object-relational mappers) [2].

The development of such applications is typically done by writing code or models, which tends to be tedious and error-prone given the repetitive scenarios that must be implemented by the developers. For example, two common scenarios consist in (i) filling a user interface with data retrieved from the server while maintaining a local cached version of these data in case of network failure and (ii) sending data to the server when the user interacts with the application while maintaining a stack of redoable actions in case of network failure.

As a solution, we introduce a data-driven approach, called End-to-end Graph Mapping (EGM), where (i) all the specialized data structures are materialized views of a larger dataset queried by end users, this dataset being stored by the GDBMS, and (ii) the whole application is modeled as a live query over these views. The query itself is a composition of dedicated *mapping operators* that automatically transforms data from one view to another, based on the view schemas and a set of mappings between these schemas. As a benefit, the main task of the developers consists into providing the data source schemas and the mappings between them. Thanks to

[1] https://www.w3.org/DOM

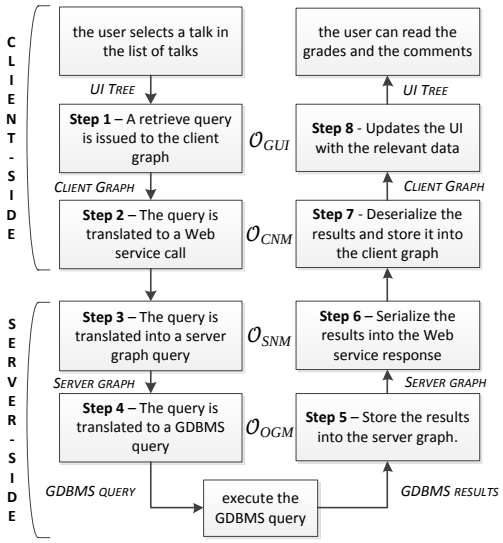

Figure 2: Typical application flow.

the schemas, the constraints defined for each data type are always ensured at each step of the application (both client- and server-side), thus reducing the risk of inadvertent errors. Thanks to the mappings, the repetitive development tasks are automated by the mapping operators that automatically transforms the data between the application parts.

The remainder of this paper is organized as follows. Section **??** discusses related work. Section 2 illustrates the core concepts and capabilities of our EGM approach, through two simple application use cases. Section 2 discusses the current EGM implementation. Section 3 concludes with our perspective for future work.

2 END-TO-END GRAPH MAPPING

Our EGM specifically targets applications composed of a client part and a server part. The server-side application includes:

- A standalone GDBMS that maintains a *full graph*, i.e., the whole set of linked data.
- A *server graph*, i.e., a graph of in-memory entities that are mapped on the actual GDBMS entities.
- Various web services for reading/updating the server graph (and, by extension, the full graph).

Similarly, the client-side application includes:

- A set of UI screens with which the user interacts.
- A *client graph*, which is a simplified version of the full graph. This client graph is maintained by the mobile application to remain usable when the network is not available (downgraded mode).

To illustrate the capabilities of our EGM, we consider a simple application use case throughout this paper. In essence, this application enables users to watch video talks and evaluate them by (i) providing comments or "likes" and (ii) by answering small sets of questions. Two scenarios are considered for covering the concepts of the EGM:

- *Scenario 1*: the user opens a talk page that displays a brief summary of the talk, the video file, the questionnaire results and the last comments.
- *Scenario 2*: once a talk page is opened, the user can provide a comment, give a "like" to the talk or answer the questionnaire.

As an example, Figure 2 illustrates Scenario 1: when a talk page is opened, (1) a retrieve query is issued to the client graph. If the data are not available locally, (2) the query is translated into a web service call, asking the server to return the data associated to this talk. The server (3) issues a retrieve query to the server graph and (4) this query is translated into the query language of the GDBMS. The GDBMS processes the query and returns a set of results (e.g., a sub-graph). The server (5) transforms these results into in-memory entities that are stored into the server graph, (6) serializes these entities using an exchange format (e.g., JSON, XML) and sends the serialized results to the client. Finally, the client (7) deserializes these results into the client graph and (8) fills the UI screen with the relevant attributes of the Talk node (e.g., the attributes that have changed).

This kind of scenario is very common in mobile and web application development and our EGM enables developers to replace each step by a well-defined mapping operator between two graphs. These steps indeed consume and produce graph-oriented data (full graph, client graph, server graph) or tree-oriented data (user interface, JSON documents) that can be mapped with each other. Our EGM defines and standardizes the mapping operators required to write such scenarios and provides reusable implementations of these operators. Once all the mapping operators are parameterized with the proper information (typically a set of mapping definitions), the application can be represented as a sequence of graph mappings between the user interface and the full graph.

The remainder of this section will describe the mapping operators introduced by our EGM for client-side (mapping the user interface to the client graph), server-side (mapping the GDBMS content to the server graph) and between the two sides.

3 CONCLUSION

In this paper we proposed the concept of End-to-End Graph Mapper: a set of mapping operators for representing web and mobile applications as (i) a multistore system composed of dedicated graph databases and (ii) live queries over this multistore system, in such a way that the graph data are mapped from one database to another. This work is still in progress and we plan to extend it in several direction.

REFERENCES

[1] Montiago Labute and Matthew Dombroski. 2014. *Review of Graph Databases for Big Data Dynamic Entity Scoring*. Technical Report. Lawrence Livermore National Laboratory.
[2] Craig Russell. 2008. Bridging the Object-Relational Divide. *Queue* 6, 3 (2008).
[3] Alessandra Toninelli, Animesh Pathak, and Valérie Issarny. 2011. Yarta: A Middleware for Managing Mobile Social Ecosystems. In *Proc of the 2011 International Conference on Advances in Grid and Pervasive Computing*. Springer.

Retour d'expérience sur l'analyse des données d'un tunnelier

Marie Le Guilly
Jean-Marc Petit
Marian Scuturici

ABSTRACT

Ce papier présente un retour d'expérience tiré d'une étude réalisée pour l'entreprise Dodin Campenon Bernard du groupe Vinci, portant sur la valorisation des données générées par un tunnelier. Les tunneliers sont d'énormes machines, ressemblant à des "petits trains" équipées de nombreux capteurs pour le guidage, le forage, l'excavation des matériaux, ... L'objectif était de déterminer si les données issues d'un tunnelier permettaient de prédire des événements sur le chantier, par exemple pour améliorer la maintenance prédictive du tunnelier et/ou réduire les coûts liés aux incidents. L'étude a duré six mois et a permis de toucher les étapes classiques de ce type de projet : identification des sources de données, intégration des données, analyse des données et validation avec les experts métiers. L'étude a montré l'intérêt des données pour l'entreprise, tout en pointant qu'une amélioration significative des résultats ne serait possible qu'en améliorant les processus menant à l'acquisition des données lors du fonctionnement d'un tunnelier.

Résumé des démonstrations

ChaseFUN: Un moteur d'Échange de Données efficace avec (et malgré) les dépendances fonctionnelles

Angela Bonifati
Ioana Ileana
Michele Linardi

ABSTRACT

Très fréquentes en pratique, les dépendances fonctionnelles sur le schéma cible restent pourtant problématiques pour les algorithmes et systèmes d'Échange de Données actuels. Les conséquences varient d'un support incomplet à un support dont la contre-partie est une baisse dramatique de performance. Nous présentons ici ChaseFUN, un moteur d'Échange de Données (ED) qui permet de sensiblement mitiger cet impact négatif, offrant ainsi un traitement efficace et non limité des dépendances fonctionnelles cible, et ceci même dans le cadre de scénarios ED complexes et de tailles considérables. ChaseFUN emploie et raffine la procédure de chase, exploitant essentiellement un ordonnancement efficace des étapes ("pas") de chase et les interactions des dépendances pour traiter de manière granulaire, paralléliser et ainsi fortement accélérer cette procédure. Les concepts et structures au cœur de notre système ont une plaisante propriété additionnelle : ils permettent de lever le voile sur un ensemble d'aspects internes souvent opaques de la chase. Ainsi, les deux atouts majeurs de ChaseFUN sont : (i) sa haute performance et facilité à passer à l'échelle ; (2) sa capacité à fournir une vue détaillée et granulaire sur le processus d'Échange de Données et la procédure de chase. Avec nos scénarios de démonstration, nous allons montrer et souligner la performance pratique de notre système et son passage à l'échelle ; de plus, nous allons offrir à l'utilisateur un regard nouveau sur la structure interne d'un scénario ED et les coulisses du déroulement de la chase. Cette démonstration a été publiée et présentée à la 20th International Conference on Extending Database Technology (EDBT) 2017.

MathMOuse: A Mathematical MOdels WarehoUSE to handle both Theoretical and Numerical Data

Cyrille Ponchateau
Ladjel Bellatreche
Carlos Ordonez
Mickael Baron

ABSTRACT

The evolution of the numeric technologies triggered an increase in the volume of data to process, along with the technical solutions to handle those volumes. Time series processing is not an exception, since they are widely used in many fields such as finance, medicine or physics. Our studied domain concerns the experimental science, in general, and automatic control in particular, which intensively uses time series data. In such domains, the numerical data comes from the observations of a physical system, usually captured using sensors. The data are then processed to find a mathematical model (usually a differential equation), which models the system dynamic behavior. Due to the volume of available data and technologies to process them, the number of the available mathematical models is increasing as well. Therefore, storing and organizing those models to ease their management and retrieval has become an essential challenge. As a solution, we propose MathMOuse, an enriched Data Warehouse structure storing mathematical models, instead of their raw numerical data (time series). From an automation scientists point of view, MathMOuse is a "query by data" system, where she/he can query theWarehouse considering the raw time series as a query parameter.During the demonstration, we first illustrate different GUI associated to MathMOuse, in terms of storing models, navigating through them, visualizing their data, etc. Then, we will focus on a use case, where a user comes with her/his raw time series data, obtained from an experiment, and checks whether a model corresponding to these data exists in the warehouse. In the case, where the model exists, the user will save time and efforts, since she/he is not obliged to elaborate the mathematical model representing her/his data.

Une infrastructure d'autocomplétion pour SPARQL générique et multi-services

Karima Rafes
Sarah Cohen-Boulakia
Serge Abiteboul

ABSTRACT

SPARQL s'est imposé comme le langage de requêtes le plus utilisé pour accéder aux masses de données RDF disponibles sur le Web. Néanmoins, rédiger une requête en SPARQL peut se révéler fastidieux, y compris pour des utilisateurs expérimentés. Cela tient souvent d'une maîtrise imparfaite par l'utilisateur des ontologies impliquées pour décrire les connaissances. Pour pallier ce problème, un nombre croissant d'éditeurs de requêtes SPARQL proposent des mécanismes d'autocomplétion. La fonctionnalité d'autocomplétion reste pourtant très limitée : elle est souvent associée à un unique champ et toujours associée à un service SPARQL fixé. Dans cet article, nous montrons comment supporter une autocomplétion générique dans un contexte de requêtes fédérées (multi-services). Nous introduisons une infrastructure permettant de proposer des complétions d'une requête en cours de rédaction en exploitant les requêtes similaires déjà posées par d'autres utilisateurs, et ce dans un contexte multi-services.Nous démontrons, au travers d'une expérimentation, la faisabilité de notre approche en nous appuyant sur un éditeur SPARQL auquel nous avons ajouté des mécanismes d'autocomplétion qui supportent une ontologie en perpétuelle évolution, ici avec la base de connaissances collaborative de Wikidata.

Strider: An Adaptive, Inference-enabled Distributed RDF Stream Processing Engine

Xiangnan Ren
Olivier Curé
Ke Li
Jeremy Lhez
Badre Belabbess
Tendry Randriamalala
Yufan Zheng
Gabriel Kepeklian

ABSTRACT

Real-time processing of data streams emanating from sensors is becoming a common task in industrial scenarios. An increasing number of processing jobs executed over such platforms are requiring reasoning mechanisms. The key implementation goal is thus to efficiently handle massive incoming data streams and support reasoning, data analytic services. Moreover, in an on-going industrial project on anomaly detection in large potable water networks, we are facing the effect of dynamically changing data and work characteristics in stream processing. The Strider system addresses these research and implementation challenges by considering scalability, fault-tolerance, high throughput and acceptable latency properties. We will demonstrate the benefits of Strider on an Internet of Things-based real world and industrial setting.

Parallelizing Query Rewriting for Key-Value Stores Under Simple Semantic Constraints

Olivier Rodriguez
Corentin Colomier
Cecilie Rivière
Reza Akbarinia
Federico Ulliana

ABSTRACT

We propose to demonstrate a system for parallelizing query rewriting over Key-Value stores in the presence of semantic constraints, as considered in the setting of ontology-mediated data integration. The constraints we consider are expressed in a native rule language for JSON records, and their purpose is to establish a unified view of data over a collection of legacy Key-Value stores. We focus on rewriting techniques for MongoDB queries allowing us to take into account the semantic constraints and parallelize the computation. During the demonstration, attendees will be able to chose queries of different complexities together with a set of rules representing semantic constraints. Then, it will be possible to see the number of rewritings generated and compare the performances of the parallel version of the algorithm with the baseline centralized one.

Résumés des articles de doctorant·e·s

Guaranteed Confidentiality and Efficiency in Crowdsourcing Platforms
Garanties de confidentialité et d'efficacité sur les plate-formes de crowdsourcing

Joris Duguépéroux
Univ. Rennes 1, IRISA, ENS Rennes
joris.dugueperoux@irisa.fr
PhD supervised by Allard Tristan and Gross-Amblard David
Work funded by the ANR CROWDGUARD project (ref. ANR-16-CE23-0004)

ABSTRACT

Crowdsourcing platforms offer the unprecedented opportunity to easily connect on-demand task providers, or requesters, and on-demand task solvers, or workers, locally or world-wide, for paid or voluntary work, and for various kinds of tasks. However, abusive behaviors from crowdsourcing platforms are frequently reported in the news or on dedicated websites, whether performed willingly or not, putting them at the epicenter of a burning societal debate, and the protection of workers in the context of crowdsourcing is often ignored in current literature. During this PhD, we will focus on ways to protect workers in crowdsourcing. We also propose here a short review of a preliminary work that has been done, to protect their privacy during the phase of assignment to tasks.

Les plate-formes de crowdsourcing offrent des opportunités sans précédent pour mettre facilement en contact des fournisseurs de tâches et des travailleurs en ligne. Cependant, des comportements problématiques sont fréquemment relevés pour ces plate-formes (espionnage des travailleurs, fuite d'information personnelle, ...). Ces comportements les mettent au centre d'un débat social, alors même que la protection des travailleurs est souvent ignorée dans la littérature actuelle. Durant ce doctorat, nous nous focaliserons sur les manières de protéger les travailleurs dans le crowdsourcing. Nous présentons ici un travail préliminaire donnant la première méthode d'affectation de tâches qui soit à la fois respectueuse de la vie privée et efficace en temps d'exécution.

1 INTRODUCTION

The goal of this PhD thesis is to design sound protection measures in the context of crowdsourcing, to prevent damage coming from the platform, while still enabling the latter to perform efficient and accurate tasks assignments. We advocate for an approach that uses confidentiality and privacy guarantees as building blocks for preventing various abusive behaviors. First, the enforcement of privacy and confidentiality guarantees will directly prevent the first kind of abuse that we consider, *i.e.,* the abusive usage of the personal or confidential information disclosed to the platform for the assignment of tasks. Second, through their obfuscation abilities, privacy and confidentiality guarantees carry the promise, in an extended form, to be also efficient for preventing a large variety of abusive behaviors (e.g, non-discrimination, or workers' independence).

During this thesis, we will (Point 1) specify relevant use-cases, extracted from real-life situations and illustrating the need to protect the crowd from various abusive behaviors from the platform. We will then (Point 2) propose secure distributed algorithms for collaboratively computing privacy-preserving versions of the information sent to the platform.

In this paper, we present the work performed during my M2 internship on point 2. We focus here on a proposition to protect privacy of workers while enabling efficient assignment to tasks. We propose to use privacy and cryptography techniques to prevent identifications. These techniques have been so far very sparsely used in topic-aware crowdsourcing: the authors of [5] propose a fully encrypted approach for the secure assignment, but appears to be unusable due to prohibitive computation costs while [1] focuses on differential privacy exclusively, from the user point of view, at the expense of the quality of results. In our approach, we aim at combining this two views in order to find an interesting compromise for these issues.

The plan of this paper is the following. First we explain the model we will use and the tools which are used to allow privacy in Section 2. Then, we will develop our current contribution in Section 3. Finally, we conclude in Section 4.

2 PROBLEM

2.1 Data Model

To allow an efficient assignment between tasks and workers, algorithms have been developed and optimized that use many private information about workers, from the expected wage to their preferences or skills.

In this work, we consider topic-aware crowdsourcing, which means we will model both tasks' requirements and workers' abilities. Workers will be modelled as both a representation of their preferences or skills, in an array of real numbers representing their degree of expertise, and individuals able to communicate with other parties. The tasks are modelled as the abilities required to be accomplished (in an array too), and the number of workers they need.

Even with no unique identifier, this precise profile of workers is likely to lead to a precise identification: for instance, knowing that

a person likes cars, can code in Java and Perl but not in C++, owns a cat, and lives in France may lead to very few candidates.

2.2 Differential Privacy

Differential privacy is currently the state-of-the-art standard when it comes to data privacy and sanitizing data. The model induced allows interactions with a database only through aggregations of results (means, sums, counts…). The main idea is to make individuals indistinguishable from each other, within a database. This is often reached by adding noise in query results in order to preserve privacy. For more information, see [2, 3].

2.3 Security model

In our approach, we will consider as attackers anyone that is not directly concerned by the assignment of a task. This means that both workers and requesters can be seen as potential attackers, but also that we do not trust any third party: the platform cannot be trusted. However, in crowdsourcing, workers and requesters have to contact each other (either to give a reward, or to receive a task), therefore, after the assignment is accomplished, parties which are directly related may learn information about each other.

In our context, attackers are considered to be *honest but curious*. This means that they will comply with the rules and algorithms we tell them to execute, but that they may also keep records of every data that they will see, and try to infer secrets and information from it.

Our scheme is considered as *secure* against honest-but-curious participants if and only if no participant can learn anything about the workers profiles that has not been encrypted or perturbed by a differentially-private mechanism. Hence, any information that attackers can access has to be either encrypted, noisy or both, so that no clear personal information is accessible.

3 CURRENT CONTRIBUTIONS

3.1 Pre-assignment

The pre-assignment phase consists in computing a partitioning the space of preferences of the workers.

To proceed with such a partitioning, we will compute a private KD-tree, in a distributed way. The main issue for this computation is to perform private sums, which is the basic building bloc. To do so, workers will first add some noise to their private values, and then encrypt them before sending them to a similar third party (trust is not required). Computation will then be performed on these encrypted profiles without revealing them thanks to homomorphic cryptography [4, 6], and the sum of the sent values will then be decrypted. Thanks to the carefully chosen noise added, this aggregated result is differentially private, so that it does not impact privacy.

Using this building step, private KD-trees can then be computed to partition the space by approaching medians thanks to histograms, without assuming any trusted third party.

3.2 Assignment and Post-assignment

Tanks to the partitioning from the pre-assignment phase taken as an input, the assignment itself can begin. The assignment phase consists into giving each task a non-empty set of the partitions in the pre-assignment phase. In this work, the pre-assignment phase gives us a sanitized count of workers for each partition. As each partition contains a set of workers, we consider that after this step, workers contained in the partitions contact the task they have been affected to, in the post-assignment phase.

At the end of the assignment phase, the contact has to be established between workers and requesters (here assimilated with the tasks they propose). To do so, we consider that each partition will be associated with an online, secure platform (accessible via TOR for instance). On this platform, each task will post a secure way to be contacted by workers, for each partition they have been associated with in the assignment phase. Note that this contact is not synonymous for accepting a task: we prefer to see it as a retractable proposition. After this contact has been made, some requesters will answer this contact and give information on the task to be done. The workers accept it or not, and send the (definitive) answer.

4 CONCLUSION

The work produced consists in the proposition of both a scheme for privacy-preserving assignment in crowdsourcing, and an instance of this scheme, along with some optimizations of it, in order to begin this PhD thesis. This scheme is separated into three main phases. The pre-assignment aims at producing a partitionned space of the proportions of workers. The assignment phase assigns areas to tasks. The post assignment ends the process by establishing contact between workers and requesters.

To continue my PhD, I aim at implementing that scheme, in order to test and measure its efficiency. In parallel, I wish to develop and formalize other forms of protections which may be useful in crowdsourcing: in terms of independence from the employer, diversity, non-discrimination, etc.

REFERENCES

[1] Louis Béziaud, Tristan Allard, and David Gross-Amblard. Lightweight Privacy-Preserving Task Assignment in Skill-Aware Crowdsourcing. In *International Conference on Database and Expert Systems Applications*, volume 10439 of *Lecture Notes in Computer Science*, pages 18 – 26, Lyon, France, August 2017.

[2] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 1–12, 2006.

[3] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[4] Oded Goldreich. Foundations of cryptography–a primer. *Foundations and Trends® in Theoretical Computer Science*, 1(1):1–116, 2005.

[5] Hiroshi Kajino. *Privacy-Preserving Crowdsourcing*. PhD thesis, 2015.

[6] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.

Une nouvelle algèbre pour SPARQL permettant l'optimisation des requêtes contenant des Property Paths

Louis Jachiet
Pierre Geneves
Nabil Layaida
Nils Gesbert

ABSTRACT

SPARQL est un langage de requêtes sur les graphes RDF standardisé par le W3C. Depuis sa version 1.1, SPARQL autorise les expressions de chemin (Property Paths) et ces expressions posent de nouveaux défis aux moteurs SPARQL. En effet, en tant qu'expressions régulières, les expressions de chemin peuvent être récursives. Or l'optimisation des requêtes récursives reste un défi tant dans le monde relationnel que dans celui du web des données. Nous introduisons dans ce travail une algèbre inspirée par l'algèbre relationnelle et par l'algèbre SPARQL ainsi qu'une traduction depuis SPARQL vers cette algèbre. Nous proposons ensuite des schémas de réécriture sur les termes de cette algèbre : étant donné une requête SPARQL on peut alors la traduire dans notre algèbre puis générer de nombreux termes équivalents qui sont alors vus comme de possible plans d'exécution de la requête SPARQL initiale. Enfin, nous concluons en montrant que nos schémas de réécritures considèrent des plans d'exécution qui ne sont pas considérés par les méthodes existantes. De surcroît, nous validons ce résultat expérimentalement : nous avons implémenté un évaluateur de requêtes SPARQL basé sur cette algèbre et mettant en place cette méthode. L'efficacité de notre prototype montre l'intérêt de notre approche.

Sampling sequential patterns with an application to the analysis of visitor trajectories

Nyoman Juniarta, Chedy Raïssi, Amedeo Napoli
Université de Lorraine, CNRS, Inria, LORIA
F-54000, Nancy, France
nyoman.juniarta@loria.fr,chedy.raissi@inria.fr,amedeo.napoli@loria.fr

ABSTRACT

In this work, we study pattern sampling in order to reduce time and space complexity in sequential pattern mining. By doing this, we can retrieve some patterns without examining the entire search space. We are studying the method that has been proposed for subgroup discovery: Monte Carlo Tree Search (MCTS). In the first part of this paper, we consider sequential pattern sampling as a single-player game based on MCTS, and define its four primary steps in each iteration. MCTS builds a tree of sequence patterns, and it focuses on the exploitation of interesting nodes and the exploration of rarely visited nodes. This ensures that the resulting patterns are diverse.

Sequential pattern mining is then involved in the improvement of a recommendation system. We work on a dataset of visitor trajectories considered as sequences. Based on sequential pattern mining, we retrieve four particular types of trajectories related to museum visitors. In this way, we can build a correspondence between sequential patterns and visitor types. These correspondence can be reused for the recommendation of trajectories that is at the heart of the CrossCult project.

KEYWORDS

Sampling, sequential pattern mining, hierarchical clustering

1 INTRODUCTION

Sequential pattern mining aims at mining interesting patterns in a database of sequences. A sequence is an ordered list $\langle s_1 s_2 \dots s_m \rangle$, where s_i is an itemset $\{i_1, \dots, i_n\}$. For example, $\langle \{a, b\}\{a, c, d\} \rangle$ is a sequence with two itemsets.

A sequence $s = \langle s_1 s_2 \dots s_m \rangle$ is a subsequence of $s' = \langle s'_1 s'_2 \dots s'_n \rangle$ if there exist indices $1 \leq i_1 < i_2 < \dots < i_m \leq n$ such that $s_j \subseteq s'_{i_j}$ for all $j = 1 \dots m$ and $m \leq n$. Therefore, a sequence $\langle \{a\}\{d\} \rangle$ is a subsequence of $\langle \{a, b\}\{a, c, d\} \rangle$, while sequence $\langle \{c\}\{d\} \rangle$ is not.

Let $\mathcal{S}$ be a sequential database. The *support* of a sequence s in $\mathcal{S}$ is the number of sequences in $\mathcal{S}$ which have s as a subsequence. The task of sequential pattern mining is to retrieve frequent sequences, i.e., whose support is larger than a user-defined threshold.

There are many existing algorithms which can retrieve all frequent sequences [2, 6]. A long sequence can have a combinatorial number of subsequences. Thus, if a long sequence is frequent, these algorithms return all of its subsequences. This leads to the retrieval of many uninteresting patterns. This issue has been studied in [5, 9, 10] by introducing the concept of closed sequence. They narrow the output by disregarding sequences which have another supersequence with the same support (hence not closed).

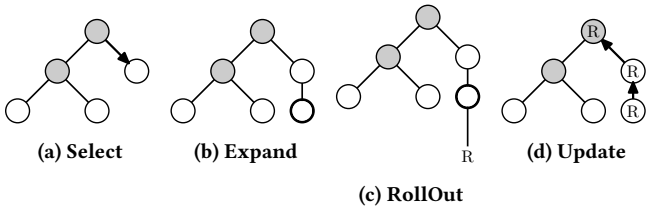

Figure 1: Four basic steps in one iteration of MCTS.

2 SAMPLING SEQUENTIAL PATTERNS

In the first step, we examined the possibility of reducing the runtime and space via a sampling method. We expect to obtain a set of frequent sequences without exhaustively examining the entire search space. MCTS is such a sampling method and was proposed by [3] to solve the problem of subgroup discovery.

In the domain of artificial intelligence, MCTS is often employed in games to build sequential decisions as a tree. It is composed by four basic steps as illustrated in Figure 1: *select*, *expand*, *rollOut*, and *update*. We extended this approach to sample sequential patterns, which is considered as a *single-turn single-player game*.

Each node in the tree represents a sequence. A child node is formed by respecting its parent's sequence, either by adding an itemset with one item or by adding an item to its parent's last itemset. For example, consider a dataset having two distinct items: a and b. A node $\langle \{b\} \rangle$ has three possible children: $\langle \{a, b\} \rangle$, $\langle \{b\}\{a\} \rangle$, and $\langle \{b\}\{b\} \rangle$. We can't add item b to create $\langle \{b, b\} \rangle$, since an itemset can't contain duplicated elements.

Select step constitutes the first part of sampling. It draws a "most-promising" node among all not-fully-expanded nodes, by selecting a node that is not fully expanded (represented as a white node in Figure 1). Starting from the root, a child node j is selected according to its $UCB1$:

$$UCB1 = \bar{X}_j + \sqrt{\frac{2 \ln n}{n_j}} \tag{1}$$

where $\bar{X}_j$ is the average reward of node j, n_j is the number of times node j has been visited, and n is number of iterations so far. This score is the Upper Confidence Bound proposed by [1], and ensures that exploration and exploitation will be equally performed. The $\bar{X}_j$

makes high-rewarded nodes to be exploited, while $\sqrt{\frac{2\ln n}{n_j}}$ prompts the exploration of rarely visited nodes.

If the node with best $UCB1$ is fully expanded, MCTS selects its child, and goes deeper until the best node is not fully expanded. Then, from that particular node, a new frequent sequence is obtained as a newly added node via *expand* step. The next two steps are then executed to update some nodes' reward, hence giving other nodes a chance to be selected.

RollOut calculates the new node's reward R. This value is obtained by randomly adding items and averaging the support of each formed sequence. Finally, MCTS does an *update* by back-propagating R through new node's parents up until the root, as shown in Figure 1d.

The MCTS method is terminated when there is no more possible node addition. This means that all frequent sequences have been added in the tree. We can limit the number of obtained sequences by setting a threshold on runtime or number of nodes. The frequent sequences are then obtained by traversing all nodes in final tree.

3 CLUSTERING SEQUENCES

Hecht Museum is an archaeological museum located in Haifa, Israel. A dataset concerning 254 of its visitors was provided for the need of the CrossCult project [7]. From those data, we attempted to cluster the visitors according to their trajectories, and subsequently mapped them into four visiting patterns: *ant*, *fish*, *butterfly*, and *grasshopper* [11].

A trajectory of a visitor is formulated as a sequence of itemsets. Each itemset corresponds to an artifact that is being visited, and contains three elements: `start_time`, `end_time`, and `artifact`. A short example of the sequence of a visitor visiting two artifacts is ⟨ {`10:00:01`, `10:05:52`, `OilLamps`}, {`10:06:11`, `10:07:43`, `JerusalemPhoto`} ⟩.

The four visiting patterns are defined according to the duration and distance between two visited artefacts, regardless the artifact's type. Accordingly, we converted the sequences such that each itemset has two elements: `duration` and `distance`. Furthermore, these two elements are discretized, such that `duration` has two possible values: `short` or `long`, while `distance` has `near` and `far`. Therefore, for all sequences, there are only four possible itemsets: {`short, near`}, {`short, far`}, {`long, near`}, and {`long, far`}.

In order to cluster the visitors according to their trajectories, a similarity measure must be defined. Distance between any two sequences can be measured by counting all of their common subsequences [4]. This measure, sim_{ACS} is formulated as:

$$sim_{ACS} = \frac{\phi_A(s_1, s_2)}{\max\{\phi_D(s_1), \phi_D(s_2)\}} \tag{2}$$

where $\phi_D(s)$ is the number of all distinct subsequences of s, while $\phi_A(s_1, s_2)$ is the number of all common subsequences between s_1 and s_2.

Given that sim_{ACT} is non-Euclidean, we chose hierarchical clustering[1] using complete linkage that works by examining the distance matrix only, without calculating any centroid.

Using 15 clusters as *hclust*'s dendrogram cut, we obtained 4 big clusters and 11 small clusters. After that, we calculated the presence of four possible itemsets in each cluster. Based on this calculation, we mapped the four clusters into four visiting patterns. An *ant* corresponds to {`long, near`} itemset, while *fish* contains {`short, near`}. The {`long, far`} itemset can be correlated with both *butterfly* and *grasshopper*, but the latter has relatively short sequence. Furthermore, within the 11 small clusters, the four possible itemsets have relatively the same support. This suggests that they correspond to visitors who frequently change their behavior while visiting the museum, and thus they contain all possible types of itemsets.

[1]We used the *hclust* method from the R software [8].

4 CONCLUSION

Sequence sampling based on MCTS can give a set of frequent sequences more efficiently in terms of space and time. A challenge arises when the number of different items is high. In the current approach, a node must be fully expanded before MCTS can select one of its children. A node is fully expanded if it has all of its possible children. With a high number of items, the number of possible children is also high. This space requirement may hinder MCTS in retrieving longer frequent sequences.

In the second part of this paper, we have demonstrated that subsequence-based clustering can be applied to characterize the behavior of a visitor. In the next step, we will incorporate the artifact's type. By inspecting the previously visited artifacts and visitor's behavior, we will be able to improve a visit recommendation system by predicting the possible next steps in a visitor trajectory.

ACKNOWLEDGMENTS

The thesis of Nyoman Juniarta is financed by the Région Grand-Est and the European project CrossCult (http://www.crosscult.eu/).

REFERENCES

[1] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.
[2] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. 2002. Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 429–435.
[3] Guillaume Bosc, Jean-François Boulicaut, Chedy Raïssi, and Mehdi Kaytoue. 2016. Anytime Discovery of a Diverse Set of Patterns with Monte Carlo Tree Search. *arXiv preprint arXiv:1609.08827* (2016).
[4] Elias Egho, Chedy Raïssi, Toon Calders, Nicolas Jay, and Amedeo Napoli. 2015. On measuring similarity for sequences of itemsets. *Data Mining and Knowledge Discovery* 29, 3 (01 May 2015), 732–764. https://doi.org/10.1007/s10618-014-0362-1
[5] Antonio Gomariz, Manuel Campos, Roque Marin, and Bart Goethals. 2013. ClaSP: an efficient algorithm for mining frequent closed sequences. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 50–61.
[6] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and MC Hsu. 2001. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering*. 215–224.
[7] Joel Lanir, Tsvi Kuflik, Eyal Dim, Alan J Wecker, and Oliviero Stock. 2013. The influence of a location-aware mobile guide on museum visitors' behavior. *Interacting with Computers* 25, 6 (2013), 443–460.
[8] R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/
[9] Jianyong Wang and Jiawei Han. 2004. BIDE: Efficient mining of frequent closed sequences. In *Proceedings of 20th International Conference on Data Engineering*. IEEE, 79–90.
[10] Xifeng Yan, Jiawei Han, and Ramin Afshar. 2003. CloSpan: Mining: Closed sequential patterns in large datasets. In *Proceedings of the 2003 SIAM International Conference on Data Mining*. SIAM, 166–177.
[11] Massimo Zancanaro, Tsvi Kuflik, Zvi Boger, Dina Goren-Bar, and Dan Goldwasser. 2007. Analyzing museum visitors' behavior patterns. In *International Conference on User Modeling*. Springer, 238–246.

Langages de requêtes interactifs pour l'exploration de données

Marie Le Guilly

ABSTRACT

Dans le contexte actuel où l'on assiste à un déluge de données, trouver des manières de naviguer efficacement dans les bases de données s'avère un challenge déterminant. En proposant des solutions basées sur des langages de requêtes connus tels que les SQL, et en empruntant des méthodes de l'apprentissage automatique, cette thèse à pour but de s'attaquer à ce challenge sous l'angle du rapprochement entre bases de données et apprentissage automatique.

Computing Schema Complements over Analytical Datasets

Rutian Liu

ABSTRACT

Cet article présente les premiers résultats pour la génération automatique de compléments de schémas pour l'exploration de données analytiques. Notre approche est fondé sur trois types de requêtes (filtre, pivot, agrégation) et la prise en compte de valeurs null dans l'identification et l'aggrégation des données.

Discovering Subsumption Axioms with Concept Annotation

Découverte d'axiomes de subsomption avec l'annotation de concepts

Pierre Monnin
LORIA (CNRS, Inria Nancy-Grand Est, Université de Lorraine)
Campus Scientifique BP 239
Vandœuvre-Lès-Nancy, France 54506
pierre.monnin@loria.fr

Amedeo Napoli
LORIA (CNRS, Inria Nancy-Grand Est, Université de Lorraine)
Campus Scientifique BP 239
Vandœuvre-Lès-Nancy, France 54506
amedeo.napoli@loria.fr

Adrien Coulet
LORIA (CNRS, Inria Nancy-Grand Est, Université de Lorraine)
Campus Scientifique BP 239
Vandœuvre-Lès-Nancy, France 54506
adrien.coulet@loria.fr

CCS CONCEPTS

• **Information systems → Data mining**; **Resource Description Framework (RDF)**; **Ontologies**; • **Computing methodologies → Ontology engineering**;

KEYWORDS

Linked Open Data, Formal Concept Analysis, Concept Annotation, ontology engineering

1 INTRODUCTION

Linked Open Data (LOD) consist of a large and growing collection of inter-domain data sets represented using Semantic Web standards including the use of RDF (Resource Description Framework) and URIs (Uniform Resource Identifiers) [1]. In LOD, resources can represent entities of the real world (*e.g.*, persons, organizations, places) and are identified with a URI. RDF statements use predicates to link resources to other resources (from the same data set or from other data sets), to literals (*e.g.*, strings, integers) or to classes of an ontology (class instantiation). Here, an ontology is a formal representation of a particular domain that consist of classes and relationships between them [3]. Among these relationships, we focus our work on the subsumption relation, stating that a class is more specific than another.

Available ontologies and LOD data sets are of various quality and completeness. In this paper, we propose to complete an ontology by discovering subsumption axioms between classes based on the regularities in a data set whose resources are linked to classes of the ontology. Particularly, we present and extend preliminary results published by the authors [6]. Formal Concept Analysis, a mathematical framework, is used to build a hierarchical structure called lattice representing the regularities between RDF resources and the predicates they are subject of. Concept Annotation is then applied on this structure to introduce ontology classes. This two-step process allows to discover subsumption axioms considering only the regularities in the data set.

2 LINKED OPEN DATA AND ONTOLOGIES

LOD are represented in the form of graphs encoded using RDF. The atomic element of an RDF graph is a triple denoted by: $\langle subject, predicate, object \rangle \in (U \cup B) \times (U \cup B) \times (U \cup B \cup L)$ where U is the set of URIs, L is the set of literals and B represent blank nodes. To illustrate our approach, we consider in this short article an abstract RDF data set (Table 1) along with an abstract ontology O (Figure 1). We denote C_O the set of classes of O. We are interested in the discovery of subsumption axioms. The subsumption relation is a transitive relation denoted by $\sqsubseteq$, where $c \sqsubseteq d$ means that every instance of c is also an instance of d. For the sake of abstraction, O uses `abstract:subClassOf` to express subsumption relations and `abstract:type` to express instantiations.

Table 1: Example of RDF triples, represented with the Turtle syntax.

r_1	abstract:type	k_1, k_2 .	r_2	$pred_3$	o_5 .
r_1	$pred_1$	o_1 .	r_3	abstract:type	k_1, k_2 .
r_1	$pred_2$	o_2 .	r_3	$pred_1$	o_6 .
r_2	abstract:type	k_1, k_2, k_4, k_5 .	r_4	abstract:type	k_1, k_2, k_5 .
r_2	$pred_1$	o_3 .	r_4	$pred_2$	o_7 .
r_2	$pred_2$	o_4 .	r_4	$pred_3$	o_8 .

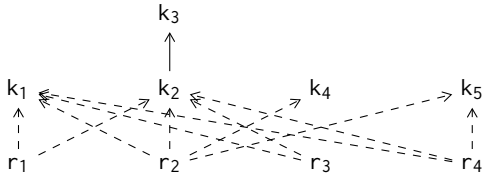

Figure 1: Example of ontology classes being instantiated by resources. Instantiations are represented using dotted arrows and subsumption relations using solid arrows. For example, r_1 instantiates k_1 and k_2, and k_2 is subsumed by k_3.

We define the *type* of a resource r as the set of classes of O that r instantiates. Formally, $\text{type}(r) = \{c \in C_O \mid \langle r, \texttt{abstract:type}, c \rangle\}$. Accordingly to the definition of the subsumption, we define the *extended type* of r as the whole set of superclasses of the classes of type(r). Formally, $\text{extdtype}(r) = \text{type}(r) \cup \{d \in C_O \mid \exists c \in \text{type}(r),\ c \sqsubseteq d\}$.

3 FORMAL CONCEPT ANALYSIS AND CONCEPT ANNOTATION

Formal Concept Analysis (FCA) is a mathematical framework whose basics can be found in [2]. It is used to build a hierarchical structure called lattice that denotes the regularities in a considered data set. Indeed, from the LOD triples in Table 1, we propose the formal

context in Table 2. It is noteworthy that only subjects and predicates are considered to build this context: a subject and a predicate are related in this context if and only if a RDF triple exists where the subject and the predicate appear together. Then, standard FCA may be applied to produce the lattice visible in Figure 2. Formal concepts (A, B) are formed by two sets: the extent A containing subjects and the intent B containing predicates. To link formal concepts with ontology classes, Concept Annotation [6] is applied to each of them. Considering a formal concept (A, B), its annotation is defined as $A^{\diamond} = \bigcap_{r \in A} \text{extdtype}(r)$. It represents the shared extended type of subjects in A. Given two formal concepts $(A_1, B_1) \leqslant (A_2, B_2)$, as $A_1 \subseteq A_2$, we have $A_2^{\diamond} \subseteq A_1^{\diamond}$. Therefore, the annotation can be depicted using an extension of the reduced labeling applied on lattices [2]. We call the *proper annotation* of a concept its annotation excluding classes appearing in the annotation of its superconcepts.

From the annotated lattice in Figure 2, subsumption axioms may then be discovered. Consider a concept (A, B), *e.g.*, concept 6, and one of its covering superconcept (E, F), *e.g.*, concept 4. As in Figure 2, we denote $A_A^{\diamond} = \{x_1, x_2, \ldots, x_p\}$ (here, $\{k_4\}$) and $E_A^{\diamond} = \{y_1, y_2, \ldots, y_q\}$ (here, $\{k_5\}$) the proper annotations of the two concepts. Then, consider two ontology classes $x_i \in A_A^{\diamond}$ and $y_j \in E_A^{\diamond}$. By definition, x_i appears in the extended type of all subjects in A and y_j appears in the extended type of all subjects in E. As $A \subseteq E$, y_j appears in the extended type of all subjects where x_i also appears. Moreover, y_j also appears in the extended type of other subjects. Thus, we consider this as the discovery of a subsumption axiom stating that x_i is subsumed by y_j (here k_4 should be subsumed by k_5). This axiom is then compared with axioms already defined in the ontology: *(i)* if $x_i \sqsubseteq y_j$ is already explicitly stated, this is a *redundant axiom*, *(ii)* if $x_i \sqsubseteq y_j$ is not already stated, but can be inferred, this is an *inferable axiom* and *(iii)* if $x_i \sqsubseteq y_j$ is neither already explicitly stated nor inferable, it is a n*ew subsumption axiom* that has been discovered. Here, the axiom $k_4 \sqsubseteq k_5$ is a new axiom.

If one considers only the covering concepts during the discovery process, this may lead to miss some interesting axioms. For example, in Figure 2, as the proper annotation of concept 2 is empty, we cannot discover axioms when considering concept 4 and its superconcept. Therefore, we propose here to discover subsumption axioms from the induced order on annotations from the lattice, obtained by considering only the annotations ordered by set inclusion following the order between formal concepts. This induced order is represented, using the reduced labeling, on the right of Figure 2. Set inclusion is read from top to bottom. Thus, subsumption axioms can be read from this reduced labeling from bottom to top, leading to discover more axioms: k_5 being subsumed by k_1, k_2 and k_3 and k_4 being subsumed by k_1, k_2 and k_3. Such axioms cannot be discovered by only considering the covering concepts.

4 PERSPECTIVES

We ran preliminary experiments on DBpedia [5], a LOD data set built from Wikipedia. It is necessary to evaluate and compare the obtained results with results of other methods. The annotation allows to take into account a third dimension (ontology classes) when describing data, in addition to the first two dimensions (RDF subjects and their predicates). Therefore, Triadic Concept Analysis [4] could be considered for comparison as it is an extension of FCA that also introduces a third dimension in the description of data. Finally, it could also be interesting to compare these two methods with results obtained using standard FCA. Indeed, because of its two-dimensional model for data description, predicates and ontology classes can only be mixed in one dimension.

Table 2: Formal context built from the RDF triples in Table 1. A cross in the table relates a subject and a predicate if and only if a RDF triple exists where the subject and the predicate appear together.

	abstract:type	$pred_1$	$pred_2$	$pred_3$
r_1	×	×	×	
r_2	×	×	×	×
r_3	×	×		
r_4	×		×	×

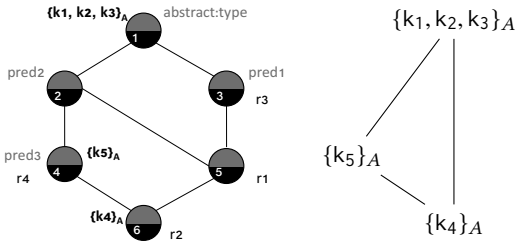

Figure 2: On the left: line diagram representing the annotated concept lattice built from the formal context in Table 2 and the ontology in Figure 1. On the right: line diagram representing the induced order on annotations. The lattice and the induced order are displayed using the reduced labeling extended to annotations. Subjects are depicted in black, predicates in grey and annotations are depicted by $\{\cdot\}_A$. Formal concepts are arbitrarily numbered from 1 to 6.

ACKNOWLEDGMENTS

This work is supported by the *PractiKPharma* project, founded by the French National Research Agency (ANR) under Grant No.: ANR-15-CE23-0028 (http://practikpharma.loria.fr/), and by the *Snowball* Inria Associate Team (http://snowflake.loria.fr/).

REFERENCES

[1] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009. Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.* 5, 3 (2009), 1–22.
[2] Bernhard Ganter and Rudolf Wille. 1999. *Formal Concept Analysis: Mathematical Foundations.* Springer.
[3] Thomas R Gruber et al. 1993. A translation approach to portable ontology specifications. *Knowledge acquisition* 5, 2 (1993), 199–220.
[4] Fritz Lehmann and Rudolf Wille. 1995. A triadic approach to formal concept analysis. *Conceptual structures: applications, implementation and theory* (1995), 32–43.
[5] Jens Lehmann and *et al.* 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195. https://doi.org/10.3233/SW-140134
[6] Pierre Monnin, Mario Lezoche, Amedeo Napoli, and Adrien Coulet. 2017. Using Formal Concept Analysis for Checking the Structure of an Ontology in LOD: The Example of DBpedia. In *Foundations of Intelligent Systems - 23rd International Symposium, ISMIS 2017, Warsaw, Poland, June 26-29, 2017, Proceedings.* 674–683. https://doi.org/10.1007/978-3-319-60438-1_66

Symmetric and Asymmetric Aggregate Function in Massively Parallel Computing

Chao Zhang
Farouk Toumani
Emmanuel Gangler

ABSTRACT

Applications of aggregation for information summary have great meanings in various fields. In big data era, processing aggregate function in parallel is drawing researchers' attention. The aim of our work is to propose a generic framework enabling to map an arbitrary aggregation into a generic algorithm and identify when it can be efficiently executed on modern large-scale data-processing systems. We describe our preliminary results regarding classes of symmetric and asymmetric aggregation that can be mapped, in a systematic way, into efficient MapReduce-style algorithms.