SIGMOD Officers, Committees, and Awardees

Chair	Vice-Chair	Secretary/Treasurer
Donald Kossmann Systems Group ETH Zürich Cab F 73 8092 Zuerich SWITZERLAND +41 44 632 29 40 <donaldk AT inf.ethz.ch>	Anastasia Ailamaki School of Computer and Communication Sciences, EPFL EPFL/IC/IIF/DIAS Station 14, CH-1015 Lausanne SWITZERLAND +41 21 693 75 64 <natassa AT epfl.ch>	Magdalena Balazinska Computer Science & Engineering University of Washington Box 352350 Seattle, WA USA +1 206-616-1069 <magda AT cs.washington.edu>

SIGMOD Executive Committee:
Donald Kossmann (Chair), Anastasia Ailamaki (Vice-Chair), Magdalena Balazinska, K. Selçuk Candan, Yanlei Diao, Curtis Dyreson, Yannis Ioannidis, Christian Jensen, and Tova Milo.

Advisory Board:
Yannis Ioannidis (Chair), Rakesh Agrawal, Phil Bernstein, Stefano Ceri, Surajit Chaudhuri, AnHai Doan, Joe Hellerstein, Michael Franklin, Laura Haas, Stratos Idreos, Tim Kraska, Renee Miller, Chris Olsten, Beng-Chin Ooi, Tamer Özsu, Sunita Sarawagi, Timos Sellis, Gerhard Weikum, John Wilkes

SIGMOD Information Director:
Curtis Dyreson, Utah State University <curtis.dyreson AT usu.edu>

Associate Information Directors:
Manfred Jeusfeld, Georgia Koutrika, Wim Martens, Mirella Moro

SIGMOD Record Editor-in-Chief:
Yanlei Diao, University of Massachusetts Amherst <yanlei AT cs.umass.edu>

SIGMOD Record Associate Editors:
Pablo Barceló, Vanessa Braganholo, Marco Brambilla, Chee Yong Chan, Rada Chirkova, Anastasios Kementsietsidis, Olga Papaemmanouil, Aditya Parameswaran, Anish Das Sarma, Alkis Simitsis, Nesime Tatbul, Marianne Winslett, and Jun Yang.

SIGMOD Conference Coordinator:
K. Selçuk Candan, Arizona State University

PODS Executive Committee:
Tova Milo (Chair), Diego Calvanse, Wenfei Fan, Martin Grohe, Rick Hull, Maurizio Lenzerini

Sister Society Liaisons:
Raghu Ramakhrishnan (SIGKDD), Yannis Ioannidis (EDBT Endowment), Christian Jensen (IEEE TKDE).

Awards Committee:
Elisa Bertino (Chair), Surajit Chaudhuri, Maurizio Lenzerini, Kartin Kersten, Umesh Dayal

Jim Gray Doctoral Dissertation Award Committee:
Tova Milo (Co-Chair), Juliana Freire (Co-Chair), Ashraf Aboulnaga, Minos Garofalakis, Chris Jermaine, Renee Miller, Aditya Parameswaran, Andy Pavlo, Kian-Lee Tan.

SIGMOD Officers, Committees, and Awardees (continued)

SIGMOD Edgar F. Codd Innovations Award

For innovative and highly significant contributions of enduring value to the development, understanding, or use of database systems and databases. Formerly known as the "SIGMOD Innovations Award", it now honors Dr. E. F. (Ted) Codd (1923 - 2003) who invented the relational data model and was responsible for the significant development of the database field as a scientific discipline. Recipients of the award are the following:

Michael Stonebraker (1992)	Jim Gray (1993)	Philip Bernstein (1994)
David DeWitt (1995)	C. Mohan (1996)	David Maier (1997)
Serge Abiteboul (1998)	Hector Garcia-Molina (1999)	Rakesh Agrawal (2000)
Rudolf Bayer (2001)	Patricia Selinger (2002)	Don Chamberlin (2003)
Ronald Fagin (2004)	Michael Carey (2005)	Jeffrey D. Ullman (2006)
Jennifer Widom (2007)	Moshe Y. Vardi (2008)	Masaru Kitsuregawa (2009)
Umeshwar Dayal (2010)	Surajit Chaudhuri (2011)	Bruce Lindsay (2012)
Stefano Ceri (2013)	Martin Kersten (2014)	Laura Haas (2015)

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services. Recipients of the award are the following:

Maria Zemankova (1992)	Gio Wiederhold (1995)	Yahiko Kambayashi (1995)
Jeffrey Ullman (1996)	Avi Silberschatz (1997)	Won Kim (1998)
Raghu Ramakrishnan (1999)	Michael Carey (2000)	Laura Haas (2000)
Daniel Rosenkrantz (2001)	Richard Snodgrass (2002)	Michael Ley (2003)
Surajit Chaudhuri (2004)	Hongjun Lu (2005)	Tamer Özsu (2006)
Hans-Jörg Schek (2007)	Klaus R. Dittrich (2008)	Beng Chin Ooi (2009)
David Lomet (2010)	Gerhard Weikum (2011)	Marianne Winslett (2012)
H.V. Jagadish (2013)	Kyu-Young Whang (2014)	Curtis Dyreson (2015)

SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to *recognize excellent research by doctoral candidates in the database field.* Recipients of the award are the following:

- **2006** *Winner*: Gerome Miklau, University of Washington. *Honorable Mentions*: Marcelo Arenas and Yanlei Diao.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley. *Honorable Mentions*: Xifeng Yan and Martin Theobald.
- **2008** *Winner*: Ariel Fuxman, University of Toronto. *Honorable Mentions*: Cong Yu and Nilesh Dalvi.
- **2009** *Winner*: Daniel Abadi, MIT. *Honorable Mentions*: Bee-Chung Chen and Ashwin Machanavajjhala.
- **2010** *Winner:* Christopher Ré, University of Washington. *Honorable Mentions*: Soumyadeb Mitra and Fabian Suchanek.
- **2011** *Winner*: Stratos Idreos, Centrum Wiskunde & Informatica. *Honorable Mentions*: Todd Green and Karl Schnaitterz.
- **2012** *Winner*: Ryan Johnson, Carnegie Mellon University. *Honorable Mention*: Bogdan Alexe.
- **2013** *Winner*: Sudipto Das, University of California, Santa Barbara. *Honorable Mention*: Herodotos Herodotou and Wenchao Zhou.
- **2014** *Winners*: Aditya Parameswaran, Stanford University, and Andy Pavlo, Brown University.
- **2015** *Winners*: Alexander Thomson, Yale University. *Honorable Mentions*: Marina Drosou, University of Ioannina and Karthik Ramachandra, IIT Bombay

A complete list of all SIGMOD Awards is available at: **http://www.sigmod.org/awards/**

[Last updated : June 30, 2015]

Editor's Notes

Welcome to the December 2015 issue of the ACM SIGMOD Record!

This issue opens with a Database Principles article by Fagin et al., which presents a relational framework for Information Extraction (IE), namely, discovering structured information in textual content. In particular, the article presents a framework, called *document spanners*, for examining the expressiveness of rule languages for IE. The spanner representation systems include regex formulas, spanner algebra, basic extraction programs, and automata. The article gives important results on the expressiveness of these representation systems. It further offers a declarative language for specifying policies for conflict resolution among different rules. The article closes by discussing other formalisms related to spanners as well as some open research questions.

The Research and Vision Articles Column features a vision article, by Kumar et al., on "Model Selection Management Systems: The Next Frontier of Advanced Analytics". This article is motivated by the observation that advanced analytics often requires running machine learning (ML) algorithms, which is an iterative process involving feature engineering, algorithm selection, and parameter tuning, collectively referred to as the *model selection* problem. Model selection, while being a highly time-consuming yet crucial task for advanced analytics, has been largely overlooked in the database community. This article envisions a new class of analytics systems called model selection management systems (MSMS), and discusses how time-tested ideas from database research offer new avenues to improving model selection.

The Surveys Column features a survey by Pournajaf et al. on "Participant Privacy in Mobile Crowd Sensing Task Management". The article focuses on participant privacy concerns and solutions in the context of task management, in contrast to privacy issues related to data collection as studied in previous work. It presents a detailed classification of task management, identifies the categories of privacy threats to participants, and provides a detailed discussion of privacy mechanisms for each type of threat. The article finally discusses ongoing research and additional challenges regarding participant privacy in Mobile Crowd Sensing task management.

The Systems and Prototypes column features a data cleaning system, "Cleanix: a Parallel Big Data Cleaning System," by Wang et al. As data cleaning is becoming a crucial task in big data analytics, Cleanix supports data cleaning at a large scale, with key features including: *scalability* on a shared-nothing cluster; *unification* of various automated data repairing tasks in a single parallel dataflow; and *usability* where users are offered with a simple and friendly graphical user interface for selecting data cleaning rules and visualization utilities for better understanding errors and fixing them.

The Distinguished Profiles column features Rick Snodgrass, Professor of Computer Science at the University of Arizona and an ACM Fellow. Rick has served as Editor-in-Chief of ACM Transactions on Database Systems, Chair of ACM SIGMOD, the ACM Pubs Board and the ACM History Committee. He has received the SIGMOD Outstanding Contributions Award and ACM Outstanding Contribution Award. Rick has been best known for his work on temporal databases. In this interview, he shared with us his thoughts on standards, branding, and his new research on "ergalics".

This issue includes three event reports. The first article reports on the Second International Workshop on Exploratory Search in Databases and the Web (ExploreDB 2015), co-located with SIGMOD 2015. The workshop included two keynote talks and six peer-reviewed research papers, which investigated a wide range of topics including explore-by-example, reformulation of database queries,

ranked search, and query personalization. The second article reports on the PhD Workshop in Information and Knowledge Management (PIKM) co-located with ACM CIKM 2014. The PIKM workshop included a regular paper track and a short paper track, both with oral and poster presentations, to increase interaction between the presenters and the audience. It also included a special track with invited talks by experienced researchers as well as a keynote speech, providing additional guidance and advice to early PhD students.

The third article in the events column reports an interesting recent study, by Benevenuto et al., on whether ACM SIG conferences have indeed promoted collaborations in a variety of research communities. More specifically, this study investigates two questions: (1) How structured are the ACM SIG conference communities? and (2) Who are the individuals responsible for connecting each ACM SIG conference community? By examining 24 ACM SIG communities and datasets from DBLP and SHINE, the article reports findings including: (1) ACM SIGMOD ranks the first among 24 communities in terms of the coverage of the largest connected component of the coauthor graph, indicating that our community is well connected in terms of collaboration; (2) a set of researchers have contributed significantly to connecting the coauthor graphs, which are well aligned with those individuals who have won research awards in their respective communities.

On behalf of the SIGMOD Record Editorial board, I hope that you all enjoy reading the December 2015 issue of the SIGMOD Record!

Your submissions to the Record are welcome via the submission site:
http://sigmod.hosting.acm.org/record

Prior to submission, please read the Editorial Policy on the SIGMOD Record's website:
http://www.sigmod.org/publications/sigmod-record/sigmod-record-editorial-policy

Yanlei Diao

December 2015

Past SIGMOD Record Editors:

Ioana Manolescu (2009-2013)
Ling Liu (2000–2004)
Arie Segev (1989–1995)
Thomas J. Cook (1981–1983)
Daniel O'Connell (1971–1973)
Alexandros Labrinidis (2007–2009)
Michael Franklin (1996–2000)
Margaret H. Dunham (1986–1988)
Douglas S. Kerr (1976-1978)
Harrison R. Morse (1969)
Mario Nascimento (2005–2007)
Jennifer Widom (1995–1996)
Jon D. Clark (1984–1985)
Randall Rustin (1974-1975)

A Relational Framework for Information Extraction

Ronald Fagin
IBM Research – Almaden
San Jose, CA, USA

Benny Kimelfeld*
Technion
Haifa, Israel

Frederick Reiss
IBM Research – Almaden
San Jose, CA, USA

Stijn Vansummeren
Universite Libre de Bruxelles (ULB)
Bruxelles, Belgium

ABSTRACT

Information Extraction commonly refers to the task of populating a relational schema, having predefined underlying semantics, from textual content. This task is pervasive in contemporary computational challenges associated with Big Data. In this article we provide an overview of our work on *document spanners*—a relational framework for Information Extraction that is inspired by rule-based systems such as IBM's SystemT.

Categories and Subject Descriptors

F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*Algebraic language theory, Classes defined by grammars or automata, Operations on languages*; F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Automata, Relations between models*; H.2.4 [**Database Management**]: Systems—*Textual databases* ; I.5.4 [**Pattern Recognition**]: Applications—*Text processing*

General Terms

Theory

Keywords

Information extraction, document spanners, regular expressions, automata, inconsistency, prioritized repairs

1. INTRODUCTION

Information Extraction (IE) refers to the task of discovering structured information in textual content. More precisely, the goal in IE is to populate a predefined relational schema that has predetermined underlying semantics, by correctly detecting the values of records in a given text document or a collection of text documents. Popular tasks in the space of IE include *named entity recognition* [29] (identify proper names in text, and classify those into a predefined set of categories such as *person* and *organization*), *relation extraction* [34] (extract tuples of entities that satisfy a predefined relationship, such as *person-organization*), *event extraction* [3] (find events of predefined types along with their key players, such as *nomination* and *nominee*), *temporal information extraction* [15,25] (associate mentions of facts with mentions of their validity period, such as *nomination-date*), and *coreference resolution* [27] (match between phrases that refer to the same entity, such as "Obama," "the President," and "him").

As a discipline, IE began with the DARPA Message Understanding Conference (MUC) in 1987 [22]. While early work in the area focused largely on military applications, this task is nowadays pervasive in a plethora of computational challenges, in particular those associated with Big Data, such as social media analysis [6], machine data analysis [21], healthcare analysis [33], semantic search [35], and customer relationship management [2]. In a typical text-analytics pipeline (e.g., [32]), the output of IE is fed into a cleaning and/or fusion component, such as an entity-resolution algorithm, that in turn produces input for a global processing phase (e.g., statistical analysis or data mining). Contemporary business models like cloud computing, along with analytics platforms like Hadoop, facilitate such data analyses for a broad range of individuals and organizations.

Most information extraction systems incorporate a notion of *rules* in a domain-specific *rule language*. These rules may define the entire extraction task, or produce features for downstream statistical models. The rules may be manually coded, or automatically learned. The choice of a rule language comprises an important part of an IE system's design. Designing such a language involves navigating a number of tradeoffs, with the most important of these being that of simplicity versus expressivity. Keeping a rule language simple pays dividends in multiple ways. A simple rule language, with relatively few language constructs and a straightforward semantics, is easier for users to understand and debug, easier for learning algorithms to train, and easier for a rule engine to execute with high throughput. But such simplicity can easily compromise the expressiveness of

*Taub Fellow, supported by the Taub Foundation

the rule language, and thus lower the quality of extraction results. Limited expressiveness of a rule language may force developers to augment the rules with custom code in a general-purpose language like Java or Python. This practice, though often necessary to achieve acceptable accuracy, makes development, maintenance, and performance tuning significantly more difficult.

This article describes our recent work on a formal framework for examining the expressivity of IE rule languages. The framework, called *document spanners*, leverages known principles of database management. The framework itself is introduced in Sections 2 and 3. In Section 4 we give results on expressiveness, and in Section 5 we discuss conflict resolution within the framework. We discuss the impact of the string-equality operator in Section 6, and conclude in Section 7.

2. DOCUMENT SPANNERS

In this section, we give some preliminary definitions and recall the formalism of *document spanners* [19].

We fix a finite alphabet Σ of *symbols*. We denote by Σ^* the set of all finite strings over Σ, and by Σ^+ the set of all finite strings of length at least one over Σ. For clarity of context, we will often refer to a string in Σ^* as a *document*. [1] A *language over* Σ is a subset of Σ^*. Let $\mathbf{d} = \sigma_1 \cdots \sigma_n \in \Sigma^*$ be a document. The length n of $\mathbf{d}$ is denoted by $|\mathbf{d}|$. A *span* identifies a substring of $\mathbf{d}$ by specifying its bounding indices. Formally, a span of $\mathbf{d}$ has the form $[i, j\rangle$, where $1 \leqslant i \leqslant j \leqslant n+1$. If $[i, j\rangle$ is a span of $\mathbf{d}$, then $\mathbf{d}_{[i,j\rangle}$ denotes the substring $\sigma_i \cdots \sigma_{j-1}$. Note that $\mathbf{d}_{[i,i\rangle}$ is the empty string, and that $\mathbf{d}_{[1,n+1\rangle}$ is $\mathbf{d}$. The more standard notation would be $[i, j)$, but we use $[i, j\rangle$ to distinguish spans from intervals. For example, $[1, 1)$ and $[2, 2)$ are both the empty interval, hence equal, but in the case of spans we have $[i, j\rangle = [i', j'\rangle$ if and only if $i = i'$ and $j = j'$ (and in particular, $[1, 1\rangle \neq [2, 2\rangle$). We denote by $\mathsf{Spans}(\mathbf{d})$ the set of all the spans of $\mathbf{d}$. Two spans $[i, j\rangle$ and $[i', j'\rangle$ of $\mathbf{d}$ are *disjoint* if $j \leqslant i'$ or $j' \leqslant i$, and they *overlap* otherwise. Finally, $[i, j\rangle$ *contains* $[i', j'\rangle$ if $i \leqslant i' \leqslant j' \leqslant j$.

EXAMPLE 2.1. In all of the examples throughout the article, we consider the example alphabet Σ which consists of the lowercase and capital letters from the English alphabet (i.e., `a`,...,`z` and `A`,...,`Z`), the comma symbol ("`,`"), and the underscore symbol ("`_`") that stands for whitespace. (We use a restricted alphabet for simplicity.) Figure 1 depicts an example document $\mathbf{d}$ in Σ^*. For ease of later reference, it also depicts the index of each character in $\mathbf{d}$. Figure 2 shows two tables containing spans of $\mathbf{d}$. Observe that the spans in the left table of Figure 2 are those that correspond to words in $\mathbf{d}$ that are names of US states (Georgia, Washington and Virginia). For example, the span $[21, 28\rangle$ corresponds to Georgia. We will further discuss the meaning of these tables later. □

[1] This is a text-only document without figures or tables.

We fix an infinite set SVars of *(span) variables*; spans may be assigned to these variables. The sets Σ^* and SVars are disjoint. For a finite set $V \subseteq \mathsf{SVars}$ of variables and a document $\mathbf{d} \in \Sigma^*$, a $(V, \mathbf{d})$-*tuple* is a mapping $\mu\colon V \to \mathsf{Spans}(\mathbf{d})$ that assigns a span of $\mathbf{d}$ to each variable in V. A $(V, \mathbf{d})$-*relation* is a set of $(V, \mathbf{d})$-tuples. A *document spanner* (or just *spanner* for short) is a function P that is associated with a finite set V of variables, denoted $\mathsf{SVars}(P)$, and that maps every document $\mathbf{d}$ to a $(V, \mathbf{d})$-relation.

EXAMPLE 2.2. Throughout the article we will define several spanners. Two of those are denoted as $[\![\rho_{\mathsf{stt}}]\!]$ and $[\![\rho_{\mathsf{loc}}]\!]$, where $\mathsf{SVars}([\![\rho_{\mathsf{stt}}]\!]) = \{x\}$ and $\mathsf{SVars}([\![\rho_{\mathsf{loc}}]\!]) = \{x_1, x_2, y\}$. Later we will explain the meaning of the $[\![\cdot]\!]$ brackets , and specify what exactly each spanner extracts from a given document. For now, the span relations (tables) in Figure 2 show the results of applying the two spanners to the document $\mathbf{d}$ of Figure 1. □

Let P be a spanner with $\mathsf{SVars}(P) = V$. Let $\mathbf{d} \in \Sigma^*$ be a document, and let $\mu \in P(\mathbf{d})$ be a $(V, \mathbf{d})$-tuple. We say that μ is *hierarchical* if for all variables $x, y \in \mathsf{SVars}(P)$ one of the following holds: *(1)* the span $\mu(x)$ contains $\mu(y)$, *(2)* the span $\mu(y)$ contains $\mu(x)$, *or (3)* the spans $\mu(x)$ and $\mu(y)$ are disjoint. As an example, the reader can verify that all the tuples in Figure 2 are hierarchical. We say that P is *hierarchical* if μ is hierarchical for all $\mathbf{d} \in \Sigma^*$ and $\mu \in P(\mathbf{d})$. Observe that for two variables x and y of a hierarchical spanner, it may be the case that, over the same document, one tuple maps x to a subspan of y, another tuple maps y to a subspan of x, and a third tuple maps x and y to disjoint spans. Finally, we say that P is *Boolean* if $\mathsf{SVars}(P)$ is empty. Note that when P is Boolean, its application to a string $\mathbf{d}$ is either the empty set (*false*) or the singleton that consists of the empty tuple (*true*).

3. SPANNER REPRESENTATION

By a *spanner representation system* we refer collectively to any manner of specifying spanners through finite objects. In this section we recall several representation systems that we have proposed and studied in previous work [18, 19]: regex formulas, spanner algebra, basic extraction programs, and automata.

3.1 Regex Formulas

Regular expressions are one of the oldest types of information extraction rule languages. Many of the systems deployed in early MUC competitions used regular

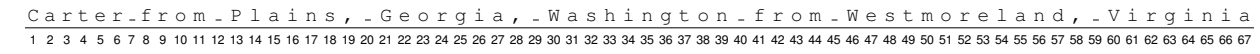


Figure 1: Document $\mathbf{d}$ in the running example

expressions over characters or token streams as their primary rule languages. A more recent example of a system with a regular expression-based rule language is the JAPE system [14], in which rules comprise regular expressions over streams of tokens, and rule evaluation is via a finite-state transducer.

A *regular expression with capture variables*, or just *variable regex* for short, is an expression in the following syntax that extends that of regular expressions:

$$\gamma \stackrel{\text{def}}{=} \varnothing \mid \epsilon \mid \sigma \mid \gamma \vee \gamma \mid \gamma \cdot \gamma \mid \gamma^* \mid x\{\gamma\} \quad (1)$$

The added alternative is $x\{\gamma\}$, where $x \in \mathsf{SVars}$. We denote by $\mathsf{SVars}(\gamma)$ the set of variables that occur in γ. We use γ^+ as abbreviations of $\gamma \cdot \gamma^*$.

A variable regex can be matched against a document in multiple ways, or more formally, there can be multiple parse trees showing that a document matches a variable regex. Each parse tree associates variables with spans. It is possible, however, that in a parse tree a variable is not associated with any span, or is associated with multiple spans. If every variable is associated with precisely one span, then the parse tree is said to be *functional*. A variable regex is called a *regex formula* if it has only functional parse trees on every input document. An example of a variable regex that is *not* a regex formula is $(x\{\texttt{a}\})^*$, because a match against `aa` assigns x to two spans. We refer to Fagin et al. [19] for the full formal definition of regex formulas. By RGX we denote the class of regex formulas. A regex formula γ is naturally viewed as representing a spanner, and by $[\![\gamma]\!]$ we denote the spanner that is represented by γ. Following are examples of spanners represented as regex formulas.

EXAMPLE 3.1. In the regex formulas of our running examples we will use the following conventions.

- `[a-z]` denotes the disjunction $\texttt{a} \vee \cdots \vee \texttt{z}$;
- `[A-Z]` denotes the disjunction $\texttt{A} \vee \cdots \vee \texttt{Z}$;
- `[a-zA-Z]` denotes `[a-z]` $\vee$ `[A-Z]`;
- Σ, by abuse of notation, denotes the regex formula recognizing all symbols in Σ, i.e., Σ denotes the disjunction `[a-zA-Z]` $\vee$ `,` $\vee$ `_`.

We now define several regex formulas that we will use throughout the article.

The following regex formula extracts tokens (which for our purposes now are simply complete words) from text. (Note that this is a simplistic extraction for the sake of presentation.)

$$\gamma_{\mathsf{tkn}} \stackrel{\text{def}}{=} \big(\epsilon \vee (\Sigma^* \cdot \texttt{\_})\big) \cdot x\{[\texttt{a-zA-Z}]^+\} \cdot \Big(\big((\texttt{,} \vee \texttt{\_}) \cdot \Sigma^*\big) \vee \epsilon\Big)$$

When applied to the document $\mathbf{d}$ of Figure 1, the resulting spans include $[1, 7\rangle$, $[8, 12\rangle$, $[13, 19\rangle$ and so on.

The following regex formula extracts spans that begin with a capital letter.

$$\gamma_{\mathsf{1cap}} \stackrel{\text{def}}{=} \Sigma^* \cdot x\{\texttt{[A-Z]} \cdot \Sigma^*\} \cdot \Sigma^*$$

When applied to the document $\mathbf{d}$ of Figure 1, the resulting spans include $[1, 7\rangle$, $[1, 3\rangle$, $[13, 19\rangle$, and so on.

The following regex formula extracts all the spans that span names of US states. For simplicity, we include just the three in Figure 1. For readability, we omit the concatenation symbol $\cdot$ between two alphabet symbols.

$$\gamma_{\mathsf{stt}} \stackrel{\text{def}}{=} \Sigma^* \cdot x\{\texttt{Georgia} \vee \texttt{Virginia} \vee \texttt{Washington}\} \cdot \Sigma^*$$

When applied to the document $\mathbf{d}$ of Figure 1, the resulting spans are $[21, 28\rangle$, $[30, 40\rangle$, and $[60, 68\rangle$.

The following regex formula extracts all the triples (x_1, x_2, y) of spans such that the string "`,_`" separates x_1 and x_2, and y is the span that starts where x_1 starts and ends where x_2 ends.

$$\gamma_{,\_} \stackrel{\text{def}}{=} \Sigma^* \cdot y\big\{x_1\{\Sigma^*\} \cdot \texttt{,\_} \cdot x_2\{\Sigma^*\}\big\} \cdot \Sigma^*$$

Let $\mathbf{d}$ be the document of Figure 1, and let V be the set $\{x_1, x_2, y\}$ of variables. The $(V, \mathbf{d})$-tuples that are obtained by applying $\gamma_{,\_}$ to $\mathbf{d}$ map (x_1, x_2, y) to triples like $([13, 19\rangle, [21, 28\rangle, [13, 28\rangle)$, and in addition, triples that do not necessarily consist of full tokens, such as the triple $([9, 19\rangle, [21, 23\rangle, [9, 23\rangle)$. □

3.2 Algebra over Spanners

Some IE systems use rule languages whose semantics derive from the relational calculus. For example, the Xlog system [28] system has a Datalog-based rule language, while SystemT [10] has a rule language based on SQL. These systems use rule engines that combine the relational algebra with automata for evaluating character-level primitives such as regular expressions. Such an algebraic runtime allows for efficient rule execution via query optimization. We can model this class of execu-

$[\![\rho_{\mathsf{stt}}]\!](\mathbf{d})$

	x
μ_1	$[21, 28\rangle$
μ_2	$[30, 40\rangle$
μ_3	$[60, 68\rangle$

$[\![\rho_{\mathsf{loc}}]\!](\mathbf{d})$

	x_1	x_2	y
μ_5	$[13, 19\rangle$	$[21, 28\rangle$	$[13, 28\rangle$
μ_4	$[21, 28\rangle$	$[30, 40\rangle$	$[21, 40\rangle$
μ_6	$[46, 58\rangle$	$[60, 68\rangle$	$[46, 68\rangle$

Figure 2: Results of spanners in the running example

tion environment by extending regex formulas with a relational algebra.

Let $\mathcal{R}$ be a representation system for spanners. Given a collection O of relational algebraic operators, we denote by $\mathcal{R}^O$ the closure of $\mathcal{R}$ under the operators of O. Here relational operators are extended pointwise to spanners. For example, consider $O = \{\bowtie\}$, where $\bowtie$ is the natural join operator. Then $\mathcal{R}^O$ consists of all spanners in $\mathcal{R}$, along with, for all spanners P_1 and P_2 definable in $\mathcal{R}$, the spanner $[\![P_1 \bowtie P_2]\!]$, which is defined by $[\![P_1 \bowtie P_2]\!](\mathbf{d}) = P_1(\mathbf{d}) \bowtie P_2(\mathbf{d})$ for all documents $\mathbf{d}$. Note in particular that the natural join here is based on span equality, not on string equality, since our relations contain spans.

We consider here three operators of positive relational algebra: union ($\cup$), projection (π), and natural join ($\bowtie$). Observe that the projection operator is parameterized by a sequence of variables from it operand spanner; that is, the operator has the form $\pi_{\mathbf{x}}$ where $\mathbf{x}$ is a sequence of variables. The standard typing rules for union and projection apply: union can only be applied to spanners P_1 and P_2 if $\mathsf{SVars}(P_1) = \mathsf{SVars}(P_2)$, and $\pi_{\mathbf{x}}$ is only applicable to spanner P if every member of $\mathbf{x}$ is in $\mathsf{SVars}(P)$. As usual, by $[\![\rho]\!]$ we denote the spanner that is represented by the algebraic expression ρ.

In the next example, we use the following notation. Let ρ be an expression in an algebra over RGX and let $\mathbf{x} = x_1, \ldots, x_n$ be a sequence of n distinct variables containing all the variables in $\mathsf{SVars}(\rho)$ (and possibly additional variables). Let $\mathbf{y} = y_1, \ldots, y_n$ be a sequence of distinct variables of the same length as $\mathbf{x}$. We denote by $\rho[\mathbf{y}/\mathbf{x}]$ the expression ρ' that is obtained from ρ by replacing every occurrence of x_i with y_i. If $\mathbf{x}$ is clear from the context, then we may write just $\rho[\mathbf{y}]$.

EXAMPLE 3.2. Using the regex formulas from Example 3.1, we define several $\mathsf{RGX}^{\{\cup,\pi,\bowtie\}}$-spanners.

- The spanner $\rho_{\mathsf{stt}} \stackrel{\text{def}}{=} \gamma_{\mathsf{tkn}} \bowtie \gamma_{\mathsf{stt}}$ extracts all the tokens that are names of US states. Note that, since $\mathsf{SVars}(\gamma_{\mathsf{tkn}}) = \mathsf{SVars}(\gamma_{\mathsf{stt}}) = \{x\}$, the natural join actually computes an intersection.
- The spanner $\rho_{\mathsf{1cap}} \stackrel{\text{def}}{=} \gamma_{\mathsf{tkn}} \bowtie \gamma_{\mathsf{1cap}}$ extracts all the tokens beginning with a capital letter.
- The spanner $\rho_{\mathsf{loc}} \stackrel{\text{def}}{=} \rho_{\mathsf{1cap}}[x_1/x] \bowtie \rho_{\mathsf{stt}}[x_2/x] \bowtie \gamma_{,\text{-}}$ extracts spans of strings including "*city*, *state*."

The results of applying the spanners $[\![\rho_{\mathsf{stt}}]\!]$ and $[\![\rho_{\mathsf{loc}}]\!]$ to the document $\mathbf{d}$ of Figure 1 are in Figure 2. Note that the right column of the right table in the figure is obtained through the spanner $\pi_y(\rho_{\mathsf{loc}})$, and the union of the left and middle columns is obtained through the spanner $\big(\pi_{x_1}(\rho_{\mathsf{loc}})\big) \cup \big(\pi_{x_2}(\rho_{\mathsf{loc}})\big)$. □

Later on, we will discuss several additional operators, including *selection*, *difference* and *complement*.

Loc	
f_1	$[13, 28\rangle$
f_2	$[21, 40\rangle$
f_3	$[46, 68\rangle$

Per	
f_4	$[1, 7\rangle$
f_5	$[13, 19\rangle$
f_6	$[21, 28\rangle$
f_7	$[30, 40\rangle$
f_8	$[46, 58\rangle$
f_9	$[60, 68\rangle$

PerLoc		
f_{10}	$[1, 7\rangle$	$[13, 28\rangle$
f_{11}	$[1, 7\rangle$	$[46, 68\rangle$
f_{12}	$[30, 40\rangle$	$[46, 68\rangle$

Figure 3: A $\mathbf{d}$-instance I over the signature of the running example

3.3 Basic Extraction Programs

In [18], we used the Datalog syntax for specifying spanners. We describe a basic form of this syntax (which we later extend) in this section.

A *signature* is a finite sequence $\mathbf{S} = \langle R_1, \ldots, R_m \rangle$ of distinct *relation symbols*, where each R_i has an arity $a_i > 0$. In this work, the *data* is a document $\mathbf{d}$, and entries in the instances of a signature are spans of $\mathbf{d}$. Formally, for a signature $\mathbf{S} = \langle R_1, \ldots, R_m \rangle$ and a document $\mathbf{d} \in \Sigma^*$, a $\mathbf{d}$-*instance* (*over* $\mathbf{S}$) is a sequence $\langle r_1, \ldots, r_m \rangle$, where each r_i is a relation of arity a_i over $\mathsf{Spans}(\mathbf{d})$; that is, r_i is a subset of $\mathsf{Spans}(\mathbf{d})^{a_i}$. A $\mathbf{d}$-*fact* (*over* $\mathbf{S}$) is an expression of the form $R(s_1, \ldots, s_a)$, where R is a relation symbol of $\mathbf{S}$ with arity a, and each s_i is a span of $\mathbf{d}$. If f is a $\mathbf{d}$-fact $R(s_1, \ldots, s_a)$ and I is a $\mathbf{d}$-instance, both over the signature $\mathbf{S}$, then we say that f is *a fact of* I if $(s_1, \ldots, s_a)$ is a tuple in the relation of I that corresponds to R. For convenience of notation, we identify a $\mathbf{d}$-instance with the set of its facts.

EXAMPLE 3.3. The signature $\mathbf{S}$ for our running example consists of three relation symbols:

- The unary relation symbol Loc stands for *location*;
- The unary relation symbol Per stands for *person*;
- The binary relation symbol PerLoc associates persons with locations.

We continue with our running example. Figure 3 shows a $\mathbf{d}$-instance over $\mathbf{S}$, where $\mathbf{d}$ is the document of Figure 1. This instance has 12 facts, and for later reference we denote them by $f_1, \ldots, f_{12}$. Note that there are quite a few mistakes in the table (e.g., the annotation of `Virginia` as a person by fact f_9); in the next section we will show how these are dealt with in the framework of this article. □

Let $\mathcal{R}$ be a spanner representation system. A *basic extraction program in* $\mathcal{R}$, or just *basic* $\mathcal{R}$*-program*, for short, is a triple $\langle \mathbf{S}, U, \varphi \rangle$, where $\mathbf{S}$ is a signature, U is a finite sequence $u_1, \ldots, u_m$ of *Horn rules*, and φ is an atomic formula over $\mathbf{S}$ (representing the result of the program). Here, an *atomic formula* φ is an expression of the form $R(x_1, \ldots, x_a)$, where R is an a-ary relation symbol in $\mathbf{S}$. A Horn rule has the form $R(y_1, \ldots, y_a) :-$

$\alpha_1 \wedge \cdots \wedge \alpha_k$, where R is a relation symbol of $\mathbf{S}$ of arity a, and each α_i is either an atomic formula over $\mathbf{S}$ or a spanner in $\mathcal{R}$. We make the requirement that each y_j occurs in at least one α_i. We denote by $\mathsf{BPR}\langle\mathcal{R}\rangle$ the class of basic $\mathcal{R}$-programs.

EXAMPLE 3.4. We now define a basic $\mathsf{RGX}^{\{\cup,\pi,\bowtie\}}$-program $\mathcal{E}$ in our running example. Intuitively, the goal of the program is to extract pairs (x, y), where x is a person and y is a location associated with x.[2] The signature is that of Example 3.3. The sequence U of rules is the following. Note that we are using the notation we established in the previous examples.

1. $\mathsf{Loc}(x) :- \rho_{\mathsf{loc}}[x]$ (see Example 3.2)
2. $\mathsf{Per}(y) :- \rho_{\mathsf{1cap}}[y]$ (see Example 3.2)
3. $\mathsf{PerLoc}(x, y) :- \mathsf{Per}(x) \wedge \mathsf{Loc}(y) \wedge \mathsf{precede}[x, y]$
4. RETURN $\mathsf{PerLoc}(x, y)$

In the above program, precede is the regex formula $\Sigma^* \cdot x\{\Sigma^*\} \cdot \Sigma^* \cdot y\{\Sigma^*\} \cdot \Sigma^*$. Hence, precede states that x terminates before y begins. □

3.4 Automata

Next, we recall a representation by means of automata. A *variable-set automaton* (or *vset-automaton*) is a tuple (Q, q_0, q_f, δ), where: Q is a finite set of *states*, $q_0 \in Q$ is an *initial state*, $q_f \in Q$ is an *accepting state*, and δ is a finite transition relation consisting of triples, each having one of the forms (q, σ, q'), (q, ϵ, q'), $(q, x\vdash, q')$ or $(q, \dashv x, q')$, where $q, q' \in Q$, $\sigma \in \Sigma$, and $x \in \mathsf{SVars}$. We denote by $\mathsf{SVars}(A)$ the set of variables that occur in the transitions of A.

Let $\mathbf{d} = \sigma_1 \cdots \sigma_n$ be a document. A *configuration* of a vset-automaton $A = (Q, q_0, q_f, \delta)$, when running on $\mathbf{d}$, is a tuple $c = (q, V, Y, i)$, where $q \in Q$ is the *current state*, $V \subseteq \mathsf{SVars}(A)$ is the set of *active variables*, $Y \subseteq \mathsf{SVars}(A)$ is the set of *available variables*, and i is an index in $\{1, \ldots, n + 1\}$. A *run* $\mathbf{c}$ of A on $\mathbf{d}$ is a sequence $c_0, \ldots, c_m$ of configurations, where $c_0 = (q_0, \varnothing, \mathsf{SVars}(A), 1)$, and for $j = 0, \ldots, m - 1$ one of the following holds for $c_j = (q_j, V_j, Y_j, i_j)$ and $c_{j+1} = (q_{j+1}, V_{j+1}, Y_{j+1}, i_{j+1})$:

1. $V_{j+1} = V_j$, $Y_{j+1} = Y_j$, and either (a) $i_{j+1} = i_j + 1$ and $(q_j, s_{i_j}, q_{j+1}) \in \delta$ (ordinary transition), or (b) $i_{j+1} = i_j$ and $(q_j, \epsilon, q_{j+1}) \in \delta$ (epsilon transition).
2. $i_{j+1} = i_j$ and for some $x \in \mathsf{SVars}(A)$, either (a) $x \in Y_j$, $V_{j+1} = V_j \cup \{x\}$, $Y_{j+1} = Y_j \backslash \{x\}$, and we have $(q_j, x\vdash, q_{j+1}) \in \delta$ (variable insert), or (b) $x \in V_j$, $V_{j+1} = V_j \backslash \{x\}$, $Y_{j+1} = Y_j$ and $(q_j, \dashv x, q_{j+1}) \in \delta$ (variable remove).

Note that in a run, each configuration (q, V, Y, i) is such that V and Y are disjoint. The run $\mathbf{c} = c_0, \ldots, c_m$ is *accepting* if $c_m = (q_f, \varnothing, \varnothing, n+1)$. We let $\mathsf{ARuns}(A, \mathbf{d})$ denote the set of all accepting runs of A on $\mathbf{d}$. If $\mathbf{c} \in \mathsf{ARuns}(A, \mathbf{d})$, then for each $x \in \mathsf{SVars}(A)$ the run $\mathbf{c}$ has a unique configuration $c_b = (q_b, V_b, Y_b, i_b)$ where x occurs in the current version of V (i.e., V_b) for the first time; and later than that $\mathbf{c}$ has a unique configuration $c_e = (q_e, V_e, Y_e, i_e)$ where x occurs in the current version of V (i.e., V_e) for the last time; the span $[i_b, i_e\rangle$ is denoted by $\mathbf{c}(x)$. By $\mu^{\mathbf{c}}$ we denote the $\mathbf{d}$-tuple that maps each variable $x \in \mathsf{SVars}(A)$ to the span $\mathbf{c}(x)$. The spanner $[\![A]\!]$ that is represented by A is the one where $\mathsf{SVars}([\![A]\!])$ is the set $\mathsf{SVars}(A)$, and where $[\![A]\!](\mathbf{d})$ is the $(\mathsf{SVars}(A), \mathbf{d})$-relation $\{\mu^{\mathbf{c}} \mid \mathbf{c} \in \mathsf{ARuns}(A, \mathbf{d})\}$. We denote by $\mathsf{VA}_{\mathsf{set}}$ the set of all variable-set automata.

As a simple example, Figure 4 depicts a vset-automaton that generates all tuples over $y_1, \ldots, y_m$

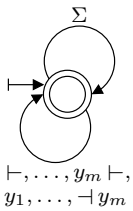


Figure 4: A vset-automaton that generates all tuples over $y_1, \ldots, y_m$

We remark that in [19] we have defined another type of automata for representing spanners, called *variable-stack* automata. We do not consider those in this article.

4. REGULAR SPANNERS AND EXPRESSIVENESS

We now give results on the expressiveness of the representation systems of the previous section. Given a representation system $\mathcal{R}$, we denote by $[\![\mathcal{R}]\!]$ the class of spanners definable by $\mathcal{R}$. The following theorem shows that several of the representation systems defined in the previous section have the same expressive power.

THEOREM 4.1. [18,19] *The following representation systems have precisely the same expressive power.*

- *The closure of regex formulas under union, projection and natural join.*
- *The basic* RGX*-programs.*
- *The vset-automata.*

That is, $[\![\mathsf{RGX}^{\{\cup,\pi,\bowtie\}}]\!] = [\![\mathsf{BPR}\langle\mathsf{RGX}\rangle]\!] = [\![\mathsf{VA}_{\mathsf{set}}]\!]$.

A spanner is *regular* if it is definable in the representation systems of Theorem 4.1. We denote by REG the set of expressions in $\mathsf{RGX}^{\{\cup,\pi,\bowtie\}}$. Hence, all of the representation systems of the theorem capture precisely

[2] In real life, such a program would of course be much more involved; here it is simplistic, for the sake of presentation.

⟦REG⟧. Note, however, that regex formulas are strictly less expressive than regular spanners. This is true, since a spanner defined by a regex formula is necessarily hierarchical. The following theorem shows that regex formulas capture *precisely* those regular spanners that are hierarchical.

THEOREM 4.2. [19] *A spanner P is definable in* RGX *if and only if P is both regular and hierarchical.*

Next, we discuss the selection operator. Let R be a k-ary string relation, and let P be a spanner. The string-selection operator ς^R is parameterized by k span variables $x_1, \ldots, x_k$ and may be written as $\varsigma^R_{x_1,\ldots,x_k}$. If P' is $\varsigma^R_{x_1,\ldots,x_k} P$, then $P'(\mathbf{d})$ is the restriction of $P(\mathbf{d})$ to those $\mathbf{d}$-tuples μ such that $(\mathbf{d}_{\mu(x_1)}, \ldots, \mathbf{d}_{\mu(x_k)}) \in R$. For example, if R is the binary relation consisting of all the pairs of strings that start with the same symbol, then $\varsigma^R_{x,y} P(\mathbf{d})$ is obtained from $P(\mathbf{d})$ by removing all the tuples γ in which the strings spanned by $\gamma(x)$ and $\gamma(y)$ start with different symbols.

A *string relation* is a relation over Σ^*. A k-ary string relation R is *recognizable* [7, 16] if it is a finite union of Cartesian products $L_1 \times \cdots \times L_k$, where each L_i is a regular language over Σ. We have the following.

THEOREM 4.3. [19] *Let R be a string relation.* RGX *is closed under the selection operator ς^R if and only if R is recognizable.*

Finally, we discuss difference and complementation. We denote by $\setminus$ the difference operator, and by $\sim$ the complement operator. Here, the *complement* of a spanner P is the spanner Q that has the same variables as P, and for every document $\mathbf{d}$, the tuples in $Q(\mathbf{d})$ are precisely those involving spans of $\mathbf{d}$ that are not in P. (Note that $Q(\mathbf{d})$ is finite since there only finitely many spans over $\mathbf{d}$.) Difference is defined as usual.

THEOREM 4.4. [19] *Regular spanners are closed under difference and complement; that is:*

$$\mathsf{REG} = \mathsf{REG}^{\{\setminus,\sim\}} = \mathsf{RGX}^{\{\cup,\pi,\bowtie,\setminus,\sim\}}$$

5. CONFLICT RESOLUTION

It is a common practice for different rules of an IE rule set to match the same region of text in different ways. Allowing this kind of overlap simplifies the task of developing and maintaining the rules if the rules are written by hand; and it simplifies the learning problem in systems that induce rules from examples. Nearly every IE rule system in use today allows for conflicting rules and provides language features for resolving these conflicts. Examples of such language features include the controls in the JAPE rule language [14] and the "consolidate" clause in SystemT's AQL [10]. The sections that follow describe our theoretical framework for a declarative language for specifying policies for conflict resolution [18].

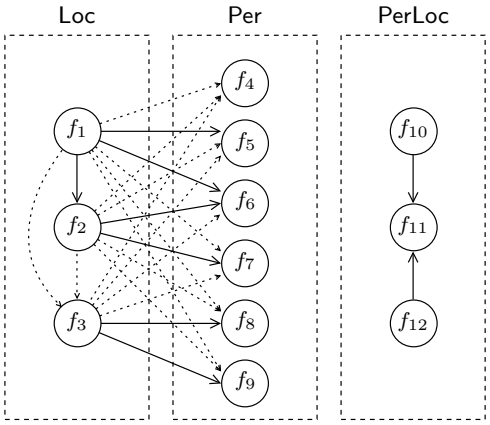


Figure 5: A conflict graph with priorities in the running example

5.1 Conflicts and Priorities

Observe that the instance of Figure 3 contains several conflicting facts. For example, f_2 represents a location, but it has a nonempty intersection with f_6 and f_7, which stand for person mentions. The database research community has established the concept of *repairs* as a mechanism for handling inconsistencies in a declarative fashion [5]. Conventionally, *denial constraints* are specified to declare sets of facts that cannot co-exist in a consistent instance. A *repair* of an inconsistent instance is a consistent subinstance that is not properly contained in any other consistent subinstance.

We adapt the concept of denial constraints to our setting. In the world of IE, the repairs are not necessarily all equal. In fact, in every example we are aware of, the developer has a clear preference as to which facts to exclude when a denial constraint is violated. Therefore, instead of the traditional repairs, we will use the notion of *prioritized repairs* of Staworko et al. [30], which extends repairing by incorporating priorities.

Let $\mathbf{S}$ be a signature, let $\mathbf{d}$ be a document, and let I be a $\mathbf{d}$-instance over $\mathbf{S}$. A *conflict hypergraph* for I is a hypergraph H over the facts of I; that is, $H = (V, E)$ where V is the set of I's facts and E is a collection of hyperedges (subsets of V). Intuitively, the hyperedges represent sets of facts that together are in conflict. A *priority relation* for I is a binary relation $\succ$ over the facts of I. If f and f' are facts of I, then $f \succ f'$ means intuitively that f is preferred to f'. A *repair* of I is a maximal subinstance of I that does not contain any hyperedge of H. To accommodate priorities in cleaning, we use the notion of *Pareto optimality* [30]: a consistent subinstance J is an *improvement* of a consistent subin-

Loc		Per		PerLoc	
f_1	$[13, 28\rangle$	f_4	$[1, 7\rangle$	f_{10}	$[1, 7\rangle$	$[13, 28\rangle$
f_3	$[46, 68\rangle$	f_7	$[30, 40\rangle$	f_{12}	$[30, 40\rangle$	$[46, 68\rangle$

Figure 6: A d-instance J_3 over the signature of the running example

stance J' if there is a fact $f \in J \backslash J'$ such that $f \succ f'$ for all $f' \in J' \backslash J$; an *optimal repair* is a consistent subinstance that has no improvement. It is easy to see that every optimal repair is also a repair in the sense of Arenas et al. [5].

EXAMPLE 5.1. Recall the instance I of our running example (Figure 3). Figure 5 shows both a conflict hypergraph (which is a graph in this case) and a priority relation over I. Specifically, the figure has two types of edges. Dotted edges (with small arrows) define priorities, where $f_i \rightarrow f_j$ denotes that $f_i \succ f_j$. Later, we shall explain the preferences (such as $f_1 \succ f_4$). Solid edges (with bigger arrows) define both conflicts and priorities: $f_i \rightarrow f_j$ denotes that $\{f_i, f_j\}$ is an edge of the conflict hypergraph, and that $f_i \succ f_j$.

Consider the following sets of facts.

$$J_1 \stackrel{\text{def}}{=} \{f_2, f_3, f_4, f_5, f_{11}\}$$
$$J_2 \stackrel{\text{def}}{=} (J_1 \cup \{f_1, f_7\}) \backslash \{f_2, f_5\}$$
$$J_3 \stackrel{\text{def}}{=} (J_2 \cup \{f_{10}, f_{12}\}) \backslash \{f_{11}\}$$

Observe that each J_i is consistent. J_2 is an improvement of J_1, since both $f_1 \succ f_2$ and $f_1 \succ f_5$ hold, and J_3 (depicted in Figure 6) is an improvement of J_2, since $f_{10} \succ f_{11}$ (and $f_{12} \succ f_{11}$). Note that J_3 is not an improvement of J_1, since no fact in J_3 is preferred to both f_2 and f_{11}. So "is an improvement of" is not transitive. The reader can verify that J_3 is an optimal repair, and in fact, the *unique* optimal repair. □

We note that another notion of optimality proposed by Staworko et al. [30] is *global optimality*, where J is an *improvement* of J' if $J \neq J'$ and for every fact $f' \in J' \backslash J$ there is a fact $f \in J \backslash J'$ such that $f \succ f'$. For the cases considered in this article, the two semantics coincide [18]. But in general, the two concepts are different. For example, in a traditional relational database with functional dependencies, optimal repair checking (i.e., given I and J, determine whether J is an optimal repair) is solvable in polynomial time in the Pareto semantics, but coNP-complete in the global semantics [17, 30].

5.2 Denial Constraints and Priority Generating Dependencies

We now discuss the syntactic declaration of conflicts and priorities. To specify a conflict hypergraph at the signature level (i.e., to specify the conflict hypergraph for every instance), we use the formalism of denial constraints. Let $\mathbf{S}$ be a signature, and let $\mathcal{R}$ be a spanner representation system. A *denial constraint* in $\mathcal{R}$ (*over* $\mathbf{S}$), or just $\mathcal{R}$*-dc* (or simply *dc*) for short, has the form

$$\forall \mathbf{x}[P \rightarrow \neg \Psi(\mathbf{x})]$$

where $\mathbf{x}$ is a sequence of variables in SVars, P is a spanner specified in $\mathcal{R}$ with all of its variables in $\mathbf{x}$, and Ψ is a conjunction of atomic formulas over $\mathbf{S}$. We usually omit the universal quantifier, and specify a dc simply by $P \rightarrow \neg \Psi(\mathbf{x})$. Semantically, $P \rightarrow \neg \Psi(\mathbf{x})$ is interpreted in the usual first-order-logic sense while viewing P as a predicate that contains all of the tuples in its output; that is, $P \rightarrow \neg \Psi(\mathbf{x})$ is satisfied in a document $\mathbf{d}$ if for every $(\mathbf{x}, \mathbf{d})$-tuple μ, if $P(\mathbf{d})$ contains the restriction of μ to $\mathsf{SVars}(P)$, then at least one of the conjuncts of Ψ must be false under μ.

EXAMPLE 5.2. We now define dcs in our running example. Recall that $\mathsf{precede}$ is a regex formula stating that x terminates before y begins. We denote by $\mathsf{disjoint}$ the regex formula $\mathsf{precede}[x, y] \vee \mathsf{precede}[y, x]$. We denote by $\mathsf{overlap}$ an expression in REG that represents the complement of $\mathsf{disjoint}$. Note that $\mathsf{overlap}$ is indeed expressible by a regular spanner, since regular spanners are closed under complement (Theorem 4.4). Finally, we denote by $\mathsf{overlap}_{\neq}$ an expression in REG that restricts the pairs in $\mathsf{overlap}$ to those (x, y) satisfying $x \neq y$ (i.e., x and y are not the same span). It is easy to verify that $\mathsf{overlap}_{\neq}$ indeed is expressible by a regular spanner.

The following dc, denoted d_{loc}, states that the spans of locations are disjoint.

$$d_{\mathsf{loc}} := \mathsf{overlap}_{\neq}[x, y] \rightarrow \neg\big(\mathsf{Loc}(x) \wedge \mathsf{Loc}(y)\big)$$

Similarly, the following dc, denoted d_{lp}, states that spans of locations are disjoint from spans of persons.

$$d_{\mathsf{lp}} := \mathsf{overlap}[x, y] \rightarrow \neg\big(\mathsf{Loc}(x) \wedge \mathsf{Per}(y)\big) \quad \square$$

To specify a priority relation $\succ$, we use what is called in [18] a *priority generating dependency*, or just *pgd* for short. Let $\mathbf{S}$ be a signature, and let $\mathcal{R}$ be a spanner representation system. A pgd in $\mathcal{R}$ (for $\mathbf{S}$) has the form $\forall \mathbf{x}[P \rightarrow (\varphi(\mathbf{x}) \succ \varphi'(\mathbf{x}))]$, where $\mathbf{x}$ is a sequence of variables in SVars, P is a spanner specified in $\mathcal{R}$ with all of its variables in $\mathbf{x}$, and φ and φ' are atomic formulas over $\mathbf{S}$. Again, we usually omit the universal quantifier and write just $P \rightarrow (\varphi(\mathbf{x}) \succ \varphi'(\mathbf{x}))$. And again, the semantics of $P \rightarrow (\varphi(\mathbf{x}) \succ \varphi'(\mathbf{x}))$ is the obvious one: for all $(\mathbf{x}, \mathbf{d})$ tuples μ, if $P(\mathbf{d})$ contains the restriction of μ to $\mathsf{SVars}(P)$, then $f \succ f'$ where f and f' are the facts that are obtained from $\varphi(\mathbf{x})$ and $\varphi'(\mathbf{x})$, respectively, by replacing every variable x with the span $\mu(x)$.

EXAMPLE 5.3. The following pgd, p_{loc}, states (using the expression $\rho[x, y]$, which is defined shortly) that for spans in the unary relation Loc, spans that start earlier are preferred, and moreover, when two spans begin together, the longer one is preferred.

$$p_{\mathsf{loc}} := \rho[x, y] \rightarrow \big(\mathsf{Loc}(x) \succ \mathsf{Loc}(y)\big)$$

Here, $\rho[x, y]$ is the following expression in REG.

$$\pi_{x,y}\Big(\big(\Sigma^* \cdot x\{z\{\epsilon\} \cdot \Sigma^*\} \cdot \Sigma^*\big) \bowtie \big(\Sigma^* \cdot z\{\epsilon\} \cdot \Sigma^+ \cdot y\{\Sigma^*\} \cdot \Sigma^*\big)\Big) \vee \big(\Sigma^* \cdot x\{y\{\Sigma^*\}\Sigma^+\} \cdot \Sigma^*\big)$$

Intuitively, the first disjunct says that x begins before y, because x begins with the empty span z, and y begins strictly after z begins. The second disjunct says that x and y begin together, but x ends strictly after y ends.

The following pgd, denoted p_{lp}, states that all the facts of Loc are preferred to all the facts of Per (e.g., because the extraction made for Loc is deemed more precise). We use the Boolean spanner true that is true on every document.

$$p_{\mathsf{lp}} := \mathsf{true} \rightarrow \big(\mathsf{Loc}(x) \succ \mathsf{Per}(y)\big) \quad \square$$

As we will discuss in Section 5.4, common resolution strategies translate into a dc and a pgd, such that the dc is binary, and the pgd defines priorities precisely on the facts that are in conflict. To refer to such a case conveniently, we write $P \rightarrow (\varphi(\mathbf{x}) \rhd \varphi'(\mathbf{x}))$ to jointly represent the dc $P \rightarrow \neg(\varphi(\mathbf{x}) \wedge \varphi'(\mathbf{x}))$ and the pgd $P \rightarrow (\varphi(\mathbf{x}) \succ \varphi'(\mathbf{x}))$. We call such a constraint a *denial pgd*.

EXAMPLE 5.4. We use $\mathsf{contains}_{\neq}[x, y]$ to denote a regex formula that produces all pairs (x, y) of spans where x strictly contains y. Let $\mathsf{enclose}[z, x, y]$ denote a specification in REG that produces all the triples (z, x, y), such that z begins where x begins and ends where y ends. For presentation's sake, we avoid the precise specification of these formulas.

The following denial pgd, denoted dp_{enc}, states that in the relation PerLoc, two facts are in conflict if the span that covers the two elements (person and location) of the first fact strictly contains that span that covers the two elements of the second; in that case, the shorter span is prioritized (since a shorter span indicates closer relationship between the person and the location).

$$\mathsf{enclose}[z, x, y] \bowtie \mathsf{enclose}[z', x', y'] \bowtie \mathsf{contains}_{\neq}[z', z] \rightarrow \mathsf{PerLoc}[x, y] \rhd \mathsf{PerLoc}[x', y'] \quad \square$$

EXAMPLE 5.5. Consider again the $\mathbf{d}$-instance I of Figure 3. The reader can verify that dcs d_{loc} and d_{lp} from Example 5.2, the pgds p_{loc} and p_{lp} in Example 5.3 and the denial pgd dp_{enc} of Example 5.4, together define the conflicts and priorities discussed in Example 5.1 (Figure 5). $\square$

Let $\mathcal{R}$ be a spanner representation system. An *extraction program in* $\mathcal{R}$, or just $\mathcal{R}$*-program* for short, is similar to a basic $\mathcal{R}$-program, except that we now allow for *cleaning rules* in addition to the Horn rules. A *cleaning rule* has the form form CLEAN$(\delta_1, \ldots, \delta_d)$, where each δ_i is a dc or a pgd in $\mathcal{R}$ (for convenience, we will also allow denial pgds).

In the program of the following example, we specify an extraction program $\langle \mathbf{S}, U, \varphi \rangle$ using only U (a sequence of rules) along with a special RETURN statement that specifies φ. We then assume that $\mathbf{S}$ consists of precisely the relation symbols that occur in the program.

EXAMPLE 5.6. We now define the REG-program $\mathcal{E}$ of our running example. Intuitively, the goal of the program is to extract pairs (x, y), where x is a person and y is a location associated with x.[3] The signature is, as usual, that of Example 3.3. The sequence U of rules is the following. Note that we are using the notation we established in the previous examples.

1. $\mathsf{Loc}(x) :- \rho_{\mathsf{loc}}[x]$ (Example 3.2)
2. $\mathsf{Per}(y) :- \rho_{\mathsf{1cap}}[y]$ (Example 3.2)
3. CLEAN$(d_{\mathsf{loc}}, d_{\mathsf{lp}}, p_{\mathsf{loc}}, p_{\mathsf{lp}})$ (Examples 5.2 and 5.3)
4. $\mathsf{PerLoc}(x, y) :- \mathsf{Per}(x) \wedge \mathsf{Loc}(y) \wedge \mathsf{precede}[x, y]$ (Example 5.2)
5. CLEAN(dp_{enc}) (Example 5.4)
6. RETURN $\mathsf{PerLoc}(x, y)$

Note that lines 1, 2 and 4 are Horn rules, whereas lines 3 and 5 are cleaning rules. $\square$

Let $\mathcal{E} = \langle \mathbf{S}, U, \varphi \rangle$ be an $\mathcal{R}$-program and let $\mathbf{d}$ be a document. Suppose that $U = \langle u_1, \ldots, u_m \rangle$. Let $\mathbf{I}_0$ be the singleton $\{I_\varnothing\}$, where $I_\varnothing$ is the empty instance over $\mathbf{S}$. For $i = 1, \ldots, m$, we denote by $\mathbf{I}_i$ the result of executing the rules $u_1, \ldots, u_i$ as we describe below. Since the cleaning operation can result in multiple instances (optimal repairs), each $\mathbf{I}_i$ is a *set* of $\mathbf{d}$-instances, rather than a single one. For $i > 0$ we define the following.

1. If u_i is the Horn rule $R(x_1, \ldots, x_a) :- \alpha_1 \wedge \cdots \wedge \alpha_k$, then $\mathbf{I}_i$ is obtained from $\mathbf{I}_{i-1}$ by adding to each $I \in \mathbf{I}_{i-1}$ all the facts (over R) that are obtained by evaluating the rule over I.

2. If u_i is the cleaning rule CLEAN$(\delta_1, \ldots, \delta_d)$, then $\mathbf{I}_i$ is obtained from $\mathbf{I}_{i-1}$ by replacing each $I \in \mathbf{I}_{i-1}$ with all the optimal repairs of I, as defined by the conflict hypergraph and priorities implied by all the δ_j.

[3] Again, our program is simplistic, for the sake of presentation.

Recall that a spanner is a function that maps a document into a $(V, \mathbf{d})$-relation (see Section 2). An extraction program acts similarly, except that a document is mapped into a set of $(V, \mathbf{d})$-relations (since it branches into multiple optimal repairs); these are all the possible resulting relations φ. For a more precise definition of the output of an extraction program, see [18]. In practice, the common case is where the extraction program produces precisely one $(V, \mathbf{d})$-relation, and then we will view the extraction program simply as a spanner.

EXAMPLE 5.7. Consider again the REG-program $\mathcal{E}$ of Example 5.6. We will now follow the steps of evaluating the program $\mathcal{E}$ on the document $\mathbf{d}$ of our running example (Figure 1). It turns out that, in this example, each $\mathbf{I}_i$ is a singleton, since every cleaning operation results in a unique optimal repair. Hence, we will treat the $\mathbf{I}_i$ as instances.

1. In $\mathbf{I}_1$, the relation Loc is as shown in Figure 3, and the other two relations are empty.
2. In $\mathbf{I}_2$, the relations Loc and Per are as shown in Figure 3, and PerLoc is empty.
3. In $\mathbf{I}_3$, the relations Loc and Per are as shown in Figure 6, and PerLoc is empty. The cleaning process is described throughout Sections 5.1 and 5.2.
4. In $\mathbf{I}_4$, the relations Loc and Per are as in $\mathbf{I}_3$, and PerLoc is as shown in Figure 3.
5. $\mathbf{I}_5$ is the instance shown in Figure 6.

The result $\mathcal{E}(\mathbf{d})$ is the $(\{x, y\}, \mathbf{d})$-relation that has two mappings: the first maps (x, y) to $([1, 7\rangle, [13, 28\rangle)$, and the second to $([30, 40\rangle, [46, 48\rangle)$. □

5.3 Cleaning in REG-Programs

We now discuss fundamental properties of extraction programs, where we focus on the class of REG-programs.

In the framework of prioritized repairs [30], the priority relation is assumed to be acyclic. We did not make such an assumption, and a pgd can indeed define a cyclic priority relation in a given program. We would like to be able to test whether acyclicity is guaranteed, but unfortunately, as the next theorem implies, no such algorithm exists for general pgds.

Let c be a cleaning rule. We say that c is *acyclic* if, for every document $\mathbf{d}$ and $\mathbf{d}$-instance I over $\mathbf{S}$, the priority relation implied by the pgds of c is acyclic.

THEOREM 5.8. [18] *Whether a pgd in* REG *is acyclic is co-recursively enumerable but not recursively enumerable. In particular, it is undecidable.*

Recall that a spanner maps a document $\mathbf{d}$ into a $(V, \mathbf{d})$-relation, for a set V of variables, while an extraction program maps $\mathbf{d}$ into a *set* of $(V, \mathbf{d})$-relations. The next property we discuss for extraction programs is that of *unambiguity*, which is the property of having a single possible world when the program is evaluated over any given document. Formally, we say that extraction program $\mathcal{E}$ is *unambiguous* if $\mathcal{E}(\mathbf{d})$ is a singleton $(V, \mathbf{d})$-relation for every document $\mathbf{d}$. We may view an unambiguous extraction program $\mathcal{E}$ simply as a specification of a spanner. The following theorem states that, unfortunately, in the presence of cleaning rules unambiguity cannot be verified for regular extraction programs.

THEOREM 5.9. [18] *Whether a* REG*-program is unambiguous is co-recursively enumerable but not recursively enumerable. In particular, it is undecidable.*

We now give a sufficient and decidable condition for unambiguity, in the case where acyclicity is guaranteed. Let I be a $\mathbf{d}$-instance over a signature $\mathbf{S}$. Let H and $\succ$ be a conflict hypergraph for I and a priority relation over I, respectively. We say that $(\succ, H)$ satisfies the *minimum property* if every hyperedge h of H contains a minimum element, that is, an element a such that $b \succ a$ for every member of h other than a. Let c be a cleaning rule. We say that c is *minimum generating* if, for every document $\mathbf{d}$ and $\mathbf{d}$-instance I over $\mathbf{S}$, for the priority relation $\succ$ and conflict hypergraph H implied by c we have that $(\succ, H)$ satisfies the minimum property. We note that for acyclic priority relations, the minimum property is less strict than the *totality* property that Staworko et al. [30] gave as a condition for unambiguity. We have the following theorem.

THEOREM 5.10. [18] *Let $\mathcal{E}$ be an $\mathcal{R}$-program for some spanner representation system $\mathcal{R}$. If every cleaning rule of $\mathcal{E}$ is acyclic and minimum generating, then $\mathcal{E}$ is unambiguous.*

In addition, we have shown that the property of being minimum generating is decidable for regular spanners.

THEOREM 5.11. [18] *Whether a given cleaning rule in* REG *is minimum generating is decidable.*

Unfortunately, the property of being acyclic is undecidable, as stated in Theorem 5.8, and so is the property of being *both* acyclic and minimum generating. Hence, as future research it is of interest to find decidable properties that imply these two properties.

Next, we address the question of whether cleaning rules increase the expressive power of extraction programs. Let $\mathcal{R}$ be a spanner representation system. A cleaning rule c defined in $\mathcal{R}$ is said to be *$\mathcal{R}$-disposable* if the following holds: for every $\mathcal{R}$-program $\mathcal{E}$ that has c as its single cleaning rule, there exists a basic (non-cleaning) $\mathcal{R}$-program that is equivalent to $\mathcal{E}$. Of course, if every cleaning rule of $\mathcal{E}$ is $\mathcal{R}$-disposable, then $\mathcal{E}$ is equivalent to a basic $\mathcal{R}$-program.

We say that a denial pgd p is *$\mathcal{R}$-disposable* if the cleaning rule that consists of only p is $\mathcal{R}$-disposable.

The following theorem implies that cleaning rules, and in fact a single acyclic denial pgd, increase the expressive power of regular extraction programs. Recall that a program that uses an acyclic denial pgd as its single cleaning rule is unambiguous (Theorem 5.10).

THEOREM 5.12. [18] *There exists an acyclic denial pgd in* REG *that is not* REG*-disposable.*

5.4 JAPE Controls

JAPE [14] is an instantiation of the Common Pattern Specification Language (CPSL) [4], a rule based framework for IE. A JAPE program (or "phase") can be viewed as an extraction program where all the relation symbols are unary. JAPE has several built-in cleaning strategies called "controls." Here, we will define these strategies in our own terminology—denial pgds.

JAPE provides four controls (in addition to the All control stating that no cleaning is to be applied). These translate to the following denial pgds. Here, R is assumed to be a unary relation in an extraction program. Under the Appelt control, $R(x) \rhd R(y)$ holds if *(1)* x and y overlap and x starts earlier than y, *or (2)* x and y start at the same position but x is longer than y. The same strategy is used is also provided by SystemT [10] (as a "consolidator"). The First control is similar to Appelt with "longer" replaced with "shorter." The Brill control is similar to Appelt, with the exclusion of option (2); that is, $R(x) \rhd R(y)$ holds if x and y overlap and x starts earlier than y. The Once control states that a single fact should remain in R (unless R is empty), which is the one that starts earliest, where a tie is broken by taking the one that ends earliest. Hence, $R(x) \rhd R(y)$ if and only if x is that remaining fact and $x \neq y$.

It is easy to show that each of the above denial pgds is acyclic, and can be expressed in REG. For example, the Appelt control is presented in Example 5.3 with R being the relation symbol Loc. While the JAPE controls can significantly simplify the programming of spanners, they do not add expressive power to regular programs, as the following theorem states.

THEOREM 5.13. [18] *Each of the denial pgds that correspond to the four JAPE controls is* REG*-disposable.*

5.5 Regular Spanners and POSIX

A regex formula γ defines a spanner by considering all possible ways that input document $\mathbf{d}$ can be matched by γ; that is, it considers all possible (functional) parse trees of γ on $\mathbf{d}$. Each such parse tree generates a new $(V, \mathbf{d})$-tuple, where $V = \mathsf{SVars}(\gamma)$, in the resulting span relation. In contrast, regular-expression pattern-matching facilities of common UNIX tools, such as `sed` and `awk`, or programming languages such as Perl, Python, and Java, do not construct all possible parse trees. Instead, they employ a *disambiguation policy* to construct only a single parse tree among the possible ones. As a result, a regex formula in these tools always yields a single $(V, \mathbf{d})$-tuple per matched input document $\mathbf{d}$ instead of multiple such tuples.[4]

In this section, we discuss the POSIX disambiguation policy [20, 23], a policy which is followed by all POSIX compliant tools such as `sed` and `awk`. Formalizations of this policy have been proposed by Vansummeren [31] and Okui and Suzuki [26], and multiple efficient algorithms for implementing the policy are known [12, 24, 26].

POSIX disambiguates as follows when matching a document $\mathbf{d}$ against regex formula γ.[5] A formal definition may be found in [26, 31]. If γ is one of $\varnothing$, ϵ, or $\sigma \in \Sigma$ then at most one parse tree exists; disambiguation is hence not necessary. If γ is a disjunction $\gamma_1 \vee \gamma_2$, then POSIX first tries to match $\mathbf{d}$ against γ_1 (recursively, using the POSIX disambiguation policy to construct a unique parse tree for this match). Only if this fails it tries to match against γ_2 (again, recursively). If, on the other hand, γ is a concatenation $\gamma_1 \cdot \gamma_2$ then POSIX first determines the longest prefix $\mathbf{d_1}$ of $\mathbf{d}$ that can be matched by γ_1 such that the corresponding suffix $\mathbf{d_2}$ of $\mathbf{d}$ can be matched by γ_2. Then, $\mathbf{d_1}$ (respectively, $\mathbf{d_2}$) is recursively matched under the POSIX disambiguation policy by γ_1 (respectively, γ_2) to construct a unique parse tree for γ. When γ is a Kleene closure δ*, there are two cases. If $\mathbf{d}$ is empty, then the entire pattern γ is taken to match $\mathbf{d}$ (irrespective of whether δ itself matches $\mathbf{d}$), and disambiguation is not necessary. If, on the other hand, $\mathbf{d}$ is nonempty, then POSIX expands γ to $\delta \cdot \delta$*. In line with the rule for concatenation, it hence first determines the longest prefix $\mathbf{d_1}$ of $\mathbf{d}$ that can be matched by δ such that the corresponding suffix $\mathbf{d_2}$ of $\mathbf{d}$ can be matched by δ*. Then, a unique parse tree for $\mathbf{d}$ against γ is constructed by matching $\mathbf{d_1}$ recursively against δ and $\mathbf{d_2}$ against δ*.

The following example illustrates the POSIX policy.

EXAMPLE 5.14. Consider $\gamma = x\{(0 \vee 01)\} \cdot y\{(1 \vee \epsilon)\}$ and $\mathbf{d} = 01$. Under the POSIX disambiguation policy, subexpression $x\{(0 \vee 01)\}$ will match as much of $\mathbf{d}$ as possible while still allowing the rest of the expression, namely $y\{(1 \vee \epsilon)\}$, to match the remainder of $\mathbf{d}$. As such, $x\{(0 \vee 01)\}$ will match $\mathbf{d}$ entirely, and $y\{(1 \vee \epsilon)\}$ will match the empty string. We hence bind x to the span $[1, 3\rangle$ and y to $[3, 3\rangle$.

[4] While our syntax $x\{\gamma\}$ for variable binding is not directly supported in these tools, it can be mimicked through the use of so-called *parenthesized expressions* and *submatch addressing*.

[5] For simplicity, we restrict ourselves here to the setting where the entire input is required to match γ. Our results naturally extend to the setting where partial matches of $\mathbf{d}$ against γ are sought.

As another example, when $\gamma = (x\{0\} \cdot y\{(1 \vee \epsilon)\}) \vee (x\{01\} \cdot y\{(1 \vee \epsilon)\})$ and $\mathbf{d} = 01$, we bind x to the span $[1, 2\rangle$ and y to the span $[2, 3\rangle$ under the POSIX disambiguation policy. □

By $\mathsf{posix}[\gamma]$ we denote the spanner represented by the regex formula γ under the POSIX disambiguation policy; this is the spanner such that $\mathsf{posix}[\gamma](\mathbf{d})$ is empty if $\mathbf{d}$ cannot be matched by γ, and consists of the unique $(V, \mathbf{d})$-tuple resulting from matching $\mathbf{d}$ against γ under the POSIX disambiguation policy otherwise.

The following theorem shows that the POSIX policy can be expressed in our cleaning framework.

THEOREM 5.15. [18] *For all regex formulas γ there exists a* REG*-program $\mathcal{E}$ such that for every document $\mathbf{d}$,*

$$\mathcal{E}(\mathbf{d}) = \{\mathsf{posix}[\gamma](\mathbf{d})\}.$$

While proving Theorem 5.15, we have observed that every cleaning rule we used in $\mathcal{E}$ is REG-disposable. Moreover, since the spanner $\mathsf{posix}[\gamma]$ is hierarchical, it follows by Theorem 4.2 that $\mathsf{posix}[\gamma]$ is itself definable in RGX by a regex formula δ. We then conclude the following theorem about POSIX, which is of interest independently of our framework.

THEOREM 5.16. [18] *For every regex formula γ, the spanner* $\mathsf{posix}[\gamma]$ *is definable in* RGX.

6. STRING EQUALITY

In this section we discuss the enrichment of regular spanners with the binary string-selection operator, denoted $\varsigma^{=}$. Given a spanner P and two variables $x, y \in \mathsf{SVars}(P)$, the application of $\varsigma^{=}_{x,y}$ selects all the tuples μ in which $\mathbf{d}_{\mu(x)} = \mathbf{d}_{\mu(y)}$. A *core spanner* [19] is a spanner definable in the algebra $\mathsf{RGX}^{\{\cup,\pi,\bowtie,\varsigma^{=}\}}$. We denote this algebra by Core.

It follows immediately from known literature on finite-state automata that core spanners have a strictly greater expressive power than regular spanners; that is, every regular spanner is a core spanner, and there are core spanners that are not regular [19]. An example of a non-regular core spanner is the following spanner, extracting all the pairs of spans with equal strings.

$$\varsigma^{=}_{x,y}\left((\Sigma^* x\{\Sigma^*\}\Sigma^*) \times (\Sigma^* y\{\Sigma^*\}\Sigma^*)\right)$$

Recall from Theorem 4.4 that regular spanners are closed under difference. The following theorem states that this is no longer the case for core spanners.

THEOREM 6.1. [19] *Assume that the alphabet Σ contains at least two symbols. Core spanners are* not *closed under difference; that is,*

$$[\![\mathsf{RGX}^{\{\cup,\pi,\bowtie,\varsigma^{=}\}}]\!] \subsetneq [\![\mathsf{RGX}^{\{\cup,\pi,\bowtie,\varsigma^{=},\setminus\}}]\!].$$

Next, we discuss the proof of Theorem 6.1. As noted in [19], the authors originally believed that the way to prove Theorem 6.1 would be to show that core spanners cannot simulate string *inequality* (i.e., select the tuples μ in which $\mathbf{d}_{\mu(x)} \neq \mathbf{d}_{\mu(y)}$). However, surprisingly, it turned out that this argument is false.

PROPOSITION 6.2. [19] *Core spanners are closed under the string-inequality operator.*

As a part of the proof of Theorem 6.1, we established the following lemma, which is of independent interest.

LEMMA 6.3. *Every core spanner is definable by an expression of the form $\pi_V SP$, where P defines a regular spanner, $V \subseteq \mathsf{SVars}(P)$, and S is a sequence of selections $\varsigma^{=}_{x,y}$ for $x, y \in \mathsf{SVars}(P)$.*

The proof of Theorem 6.1 then completes as follows. An easy observation is that core spanners are closed under the substring-of selection operator (i.e., select the tuples μ in which $\mathbf{d}_{\mu(x)}$ is a substring of $\mathbf{d}_{\mu(y)}$). But using Lemma 6.3 we have proved the following theorem.

THEOREM 6.4. [19] *Assume that the alphabet Σ contains at least two symbols. Core spanners are not closed under the* not-a-substring-of *binary operator.*

We complete this section with the following theorem, which implies that disposability of the JAPE controls no longer holds in the case of core spanner.

THEOREM 6.5. [18] *None of the JAPE denial pgds is* Core*-disposable.*

7. CONCLUDING REMARKS

We conclude this paper by some observations relating spanners to other formalisms, and some open questions.

Many practical regular expression pattern matching engines (such as the ones found in `sed`, `awk`, Perl, Java and Python) support a feature called *backreferences*. Using this feature, variables can not only bind to a substring during matching, but can also be used to repeat a previously matched substring. Regular expressions that have this feature are called *extended regular expressions* (xregex for short) [1, 8, 9]. It is known that xregex can recognize non-regular languages, such as $\{\mathbf{ss} \mid \mathbf{s} \in \Sigma^*\}$. Note that this language can be recognized by a Boolean core spanner. In [19] we established that xregex can also recognize languages that are not recognizable by any Boolean core spanner. The reverse question, whether Boolean core spanners can recognize languages that are not recognizable by any xregex, is still open.

Various languages for querying semi-structured and graph databases are based on regular expressions. A simple form of such queries are the *regular path queries*

(*RPQs*) that are applied to directed graphs with labeled edges [11, 13]. An RPQ identifies node pairs connected by a path such that the word formed by the edge labels belongs to a specified regular language. A *conjunctive regular path query* (*CRPQ*) applies conjunction and existential quantification (over nodes) to RPQs [11], and has been the subject of much investigation.

Superficially speaking, spanners and CRPQs are inherently different concepts: spanners operate on *strings* while CRPQs operate on *graphs* (directed, edge-labeled graphs); and the variables in the spanner world represent *spans*, while those in the CRPQ world represent *nodes*. However, it is possible to adjust CRPQs to represent spanners [19]. In terms of the data model, a string can be viewed as a special case of a graph, namely a simple path. Formally, given a string $\mathbf{s} = \sigma_1 \cdots \sigma_n$, we denote by $p(\mathbf{d})$ the simple path $1 \to 2 \to \cdots \to n+1$ (with the natural numbers $1, \ldots, n+1$ as nodes), where for $i = 1, \ldots, n$ the label of the edge $i \to i+1$ is σ_i. For technical reasons, it is necessary to mark the begin node 1 and end node n in this simple path with the two loops $1 \to 1$ and $(n+1) \to (n+1)$, labeled with new labels $\triangleright$ and $\triangleleft$ (not in the alphabet Σ). On this so-called *marked path*, a CRPQ can define a spanner over a set of span variables V by introducing, for each $x \in V$, two CRPQ variables: one that will indicate the start position of the span matched by x and one that indicates the end position of the span matched by x. Using this representation of spanners through CRPQ, one can show that $[\![\mathsf{REG}]\!]$ is exactly captured by unions of CRPQs, while $[\![\mathsf{Core}]\!]$ is exactly captured by unions of CRPQs extended with string equality [19].

As we have seen in this article, we have identified several representation systems for regular and core spanners that are equivalent in expressive power. While our proofs of these equivalences describe effective translations between the representation systems, it would be interesting to study the inherent complexity of these translations in order to establish their relative succinctness. A second question that deserves further attention is the complexity of evaluating spanners expressed in the various representation systems.

8. REFERENCES

[1] A. V. Aho. Algorithms for finding patterns in strings. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A)*, pages 255–300. North Holland, 1990.

[2] J. Ajmera, H.-I. Ahn, M. Nagarajan, A. Verma, D. Contractor, S. Dill, and M. Denesuk. A CRM system for social media: challenges and experiences. In *WWW*, pages 49–58, 2013.

[3] C. Aone and M. Ramos-Santacruz. Rees: A large-scale relation and event extraction system. In *ANLP*, pages 76–83, 2000.

[4] D. E. Appelt and B. Onyshkevych. The common pattern specification language. In *Proceedings of the TIPSTER Text Program: Phase III*, pages 23–30, Baltimore, Maryland, USA, 1998. Association for Computational Linguistics.

[5] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *PODS*, pages 68–79. ACM, 1999.

[6] E. Benson, A. Haghighi, and R. Barzilay. Event discovery in social media feeds. In *ACL*, pages 389–398. The Association for Computer Linguistics, 2011.

[7] J. Berstel. *Transductions and Context-Free Languages*. Teubner Studienbücher, Stuttgart, 1979.

[8] C. Câmpeanu, K. Salomaa, and S. Yu. A formal study of practical regular expressions. *Int. J. Found. Comput. Sci.*, 14(6):1007–1018, 2003.

[9] B. Carle and P. Narendran. On extended regular expressions. In *LATA 2009*, volume 5457 of *Lecture Notes in Computer Science*, pages 279–289, 2009.

[10] L. Chiticariu, R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, and S. Vaithyanathan. SystemT: An algebraic approach to declarative information extraction. In *ACL*, pages 128–137. The Association for Computer Linguistics, 2010.

[11] M. P. Consens and A. O. Mendelzon. Graphlog: a visual formalism for real life recursion. In *PODS*, pages 404–416. ACM, 1990.

[12] R. Cox. Regular expression matching: the virtual machine approach. digression: Posix submatching, December 2009. http://swtch.com/ rsc/regexp/regexp2.html.

[13] I. F. Cruz, A. O. Mendelzon, and P. T. Wood. A graphical query language supporting recursion. In *SIGMOD Conference*, pages 323–330. ACM, 1987.

[14] H. Cunningham, D. Maynard, and V. Tablan. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November 2000.

[15] M. Dylla, I. Miliaraki, and M. Theobald. A temporal-probabilistic database model for information extraction. *PVLDB*, 6(14):1810–1821, 2013.

[16] C. C. Elgot and J. E. Mezei. On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9:47–68, 1965.

[17] R. Fagin, B. Kimelfeld, and P. G. Kolaitis. Dichotomies in the complexity of preferred repairs. In *PODS*, pages 3–15. ACM, 2015.

[18] R. Fagin, B. Kimelfeld, F. Reiss, and S. Vansummeren. Cleaning inconsistencies in information extraction via prioritized repairs. In *PODS*, pages 164–175, Snowbird, Utah, 2014. ACM.

[19] R. Fagin, B. Kimelfeld, F. Reiss, and S. Vansummeren. Document spanners: A formal approach to information extraction. *J. ACM*, 62(2):12, 2015. Preliminary version appeared in PODS 2013.

[20] G. Fowler. An interpreation of the posix regex standard (2003), 2003. http://gsf.cococlyde.org/download/re-interpretation.tgz.

[21] Q. Fu, J.-G. Lou, Y. Wang, and J. Li. Execution anomaly detection in distributed systems through unstructured log analysis. In *ICDM*, pages 149–158. IEEE Computer Society, 2009.

[22] R. Grishman and B. Sundheim. Message understanding conference- 6: A brief history. In *COLING*, pages 466–471, 1996.

[23] Institute of Electrical and Electronic Engineers and the Open group. The open group base specifications issue 7, 2013. IEEE Std 1003.1, 2013 Edition.

[24] V. Laurikari. Efficient submatch addressing for regular expressions. Master's thesis, Helsinki University of Technology, 2001.

[25] X. Ling and D. S. Weld. Temporal information extraction. In *AAAI*. AAAI Press, 2010.

[26] S. Okui and T. Suzuki. Disambiguation in regular expression matching via position automata with augmented transitions. In M. Domaratzki and K. Salomaa, editors, *CIAA*, volume 6482 of *Lecture Notes in Computer Science*, pages 231–240. Springer, 2010.

[27] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. D. Manning. A multi-pass sieve for coreference resolution. In *EMNLP*, pages 492–501. ACL, 2010.

[28] W. Shen, A. Doan, J. F. Naughton, and R. Ramakrishnan. Declarative information extraction using datalog with embedded extraction predicates. In *VLDB*, pages 1033–1044. ACM, 2007.

[29] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272, 1999.

[30] S. Staworko, J. Chomicki, and J. Marcinkowski. Prioritized repairing and consistent query answering in relational databases. *Ann. Math. Artif. Intell.*, 64(2-3):209–246, 2012.

[31] S. Vansummeren. Type inference for unique pattern matching. *ACM Trans. Program. Lang. Syst.*, 28(3):389–428, 2006.

[32] R. Wisnesky, M. A. Hernández, and L. Popa. Mapping polymorphism. In *ICDT*, ACM International Conference Proceeding Series, pages 196–208. ACM, 2010.

[33] H. Xu, S. P. Stenner, S. Doan, K. B. Johnson, L. R. Waitman, and J. C. Denny. Application of information technology: Medex: a medication information extraction system for clinical narratives. *JAMIA*, 17(1):19–24, 2010.

[34] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106, 2003.

[35] H. Zhu, S. Raghavan, S. Vaithyanathan, and A. Löser. Navigating the intranet with high precision. In *WWW*, pages 491–500. ACM, 2007.

Model Selection Management Systems: The Next Frontier of Advanced Analytics

Arun Kumar[†] Robert McCann[‡] Jeffrey Naughton[†] Jignesh M. Patel[†]
[†]University of Wisconsin-Madison [‡]Microsoft
[†]{arun, naughton, jignesh}@cs.wisc.edu, [‡]robert.mccann@microsoft.com

ABSTRACT

Advanced analytics is a booming area in both industry and academia. Several projects aim to implement machine learning (ML) algorithms efficiently. But three key challenging and iterative practical tasks in using ML – *feature engineering*, *algorithm selection*, and *parameter tuning*, collectively called *model selection* – have largely been overlooked by the data management community, even though these are often the most time-consuming tasks for analysts. To make the iterative process of model selection easier and faster, we envision a unifying abstract framework that acts as a basis for a new class of analytics systems that we call *model selection management systems* (MSMS). We discuss how time-tested ideas from database research offer new avenues to improving model selection, and outline how MSMSs are a new frontier for interesting and impactful data management research.

1. INTRODUCTION

The data management community has produced successful systems that implement machine learning (ML) techniques efficiently, often over data management platforms [2, 6, 8, 11, 19]. But the process of building ML models for data applications is seldom a one-shot "slam dunk." Analysts face major practical bottlenecks in *using* ML that slow down the analytics lifecycle [3]. To understand these bottlenecks, we spoke with analysts at several enterprise and Web companies. Unanimously, they mentioned that choosing the right *features* and appropriately tuned ML models were among their top concerns. Other recent studies have produced similar findings [4, 5, 12]. In this paper, we focus on a related set of challenging practical tasks in using ML for data-driven applications: *feature engineering* (FE), in which the analyst chooses the features to use; *algorithm selection* (AS), in which the analyst picks an ML algorithm; and *parameter tuning* (PT), in which the analyst tunes ML algorithm parameters. These tasks, collectively called *model selection*, lie at the heart of advanced analytics.

Model Selection. Broadly defined, model selection is the process of building a precise prediction function to make "satisfactorily" accurate predictions about a data-generating process using data generated by the same process [10]. In this paper, we explain how viewing model selection from a data management standpoint can improve the process. To this end, we envision a *unifying framework* that lays a foundation for a new class of analytics systems: *model selection management systems*.

Model Selection Triple. A large body of work in ML focuses on various theoretical aspects of model selection [10]. But from a practical perspective, we found that analysts typically use an *iterative exploratory process*. While the process varies across analysts, we observed that the core object of their exploration is identical – an object we call the *model selection triple* (MST). It has three components: an FE "option" (loosely defined, a sequence of computation operations) that fixes the feature set that represents the data, an AS option that fixes the ML algorithm, and a PT option that fixes the parameter choices conditioned on the AS option. Model selection is iterative and exploratory because the space of MSTs is usually infinite, and it is generally impossible for analysts to know a priori which MST will yield satisfactory accuracy and/or insights.

Three-Phase Iteration. We divide an iteration into three phases, as shown in Figure 1(A). (1) *Steering*: the analyst decides on an MST and specifies it in an ML-related language or GUI such as R, Scala, SAS, or Weka. For example, suppose she has structured data; she might decide to use all features (FE option), and build a decision tree (AS option) with

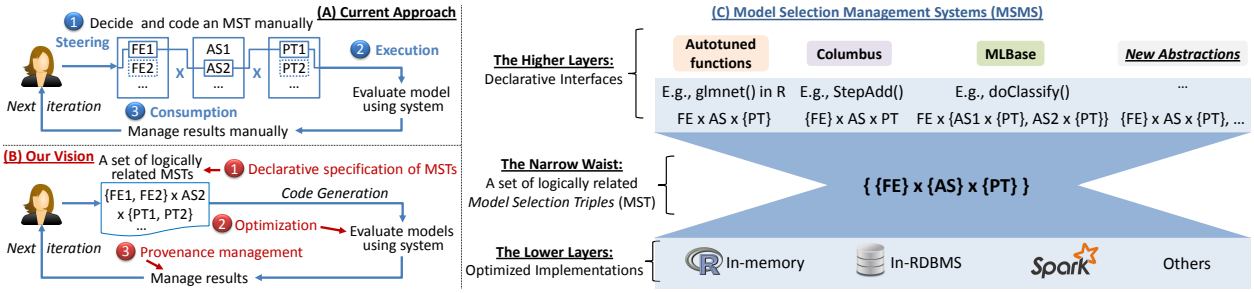


Figure 1: Model Selection. (A) The current dominant approach: the analyst chooses one combination of options for feature engineering (FE), algorithm selection (AS), and parameter tuning (PT); we call it a Model Selection Triple (MST). She iterates by modifying the MST, e.g., altering a parameter, or dropping a feature. (B) Our vision: she groups *logically related* MSTs, while the system optimizes the computations and helps manage results across iterations. (C) MSTs act as a unifying abstraction (a "narrow waist") for a new class of analytics systems that we call Model Selection Management Systems (MSMS).

a fixed tree height (PT option). (2) *Execution*: the system executes the MST to build and evaluate the ML model, typically on top of a data management platform, e.g., an RDBMS or Spark. (3) *Consumption*: the analyst assesses the results to decide upon the MST for the next iteration, or stops the process. For example, if the tree is too big, she might reduce the height (PT option changed), or drop some features (FE option changed). Even such minor changes in MSTs can cause major changes in accuracy and interpretability, and it is generally hard to anticipate such effects. Thus, analysts evaluate several MSTs using an iterative process.

Alas, most existing ML systems (e.g., [2, 6, 8, 11, 19]) force analysts to explore one MST per iteration. This overburdens the analyst, since she has to perform more iterations. Also, since the system is ignorant of the relationship between the MSTs explored, opportunities to speed up Execution are lost, while Consumption becomes more manual, which causes more pain for analysts. *Our vision aims to mitigate these issues by providing more systems support to improve the effectiveness of analysts as well as the efficiency of the iterative process of model selection.*

Automation Spectrum. A natural first step is to automate some of the analyst's work by providing a more holistic view of the process to the system. For example, the system can evaluate decision trees with all possible heights (PT options), or all subsets of features (FE options). Clearly, such naive brute-force automation might be prohibitively expensive, while the analyst's expertise might be wastefully ignored. On the other hand, if the system hardcodes only a few sets of MSTs, analysts might deem it too restrictive to be useful. Ideally, what we want is the flexibility to cover a wide *spectrum of automation*, in which the analyst controls exactly how much automation she desires for her application. This would enable analysts to still use their expertise during Steering, but push much of the "heavy lifting" to the system during Execution and Consumption.

1.1 Our Vision: MSMS to Manage MSTs

We envision a unifying framework that enables analysts to easily explore *a set of logically related MSTs per iteration* rather than just one MST per iteration. The analyst's expertise is useful in deciding which MSTs are grouped. Figure 1(B) illustrates our vision. In the figure, the analyst groups multiple values of tree height and feature subsets. As we will explain shortly, this ability to handle a logically related set of MSTs all at once is a simple but powerful unifying abstraction for a new class of analytics systems that aim to support the iterative process of model selection. We call such systems *model selection management systems* (MSMS).

Iteration in an MSMS. MSTs are too low-level for analysts to enumerate. Thus, an MSMS should provide a higher level of abstraction for specifying MSTs. A trivial way is to use *for* loops. But our vision goes deeper to exploit the full potential of our idea, by drawing inspiration from the RDBMS philosophy of handling "queries." By repurposing three key ideas from the database literature, an MSMS can make model selection significantly easier and faster. (1) Steering: an MSMS should offer a *framework of declarative operations* that enable analysts to easily group logically related MSTs. For example, the analyst can just "declare" the set of tree heights and feature subsets (projections). The

system generates lower-level code to implicitly enumerate the MSTs encoded by the declarative operations. (2) Execution: an MSMS should include *optimization* techniques to reduce the runtime per iteration by exploiting the set-oriented nature of specifying MSTs. For example, the system could share computations across different parameters or share intermediate materialized data for different feature sets. (3) Consumption: an MSMS should offer *provenance management* so that the system can help the analyst manage results and help with optimization. For example, the analyst can inspect the results using standard queries to help steer the next iteration, while the system can track intermediate data and models for reuse. *Overall, an MSMS that is designed based on our unifying framework can reduce both the number of iterations (by improving Steering and Consumption) and the time per iteration (by improving Execution).*

Unifying Narrow Waist. Interestingly, our framework subsumes many prior abstractions that can be seen as basically restricted MSMS interfaces in hindsight, as shown in Figure 1(C). In a sense, our abstraction acts as a "narrow waist" for MSMSs.[1] An MSMS stacks higher layers of abstraction (declarative interfaces) to make it *easier to specify sets of MSTs* and lower layers of optimized implementations to *exploit the set-oriented nature of specifying MSTs*. We elaborate with three key examples: (1) R provides many autotuned functions, e.g., *glmnet* for linear models. These can be viewed as declarative operations to explore multiple PT options, while fixing FE and AS options. (2) Columbus [18] offers declarative operations to explore a set of feature subsets simultaneously. This can be viewed as exploring multiple FE options, while fixing AS and PT options. (3) MLBase [13] fully automates algorithm selection and parameter tuning by hardcoding a small set of ML techniques and tuning heuristics (unlike our vision of handling a wide spectrum of automation). This can be viewed as exploring multiple combinations of AS and PT options, while fixing the FE option. *Our vision is the distillation of the common thread across such abstractions, and lays a principled foundation for the design and evaluation of model selection management systems.*

Towards a Unified MSMS. The natural next step is to build a unified MSMS that exposes the full power of our abstraction rather than supporting FE, AS, and PT in a piecemeal fashion. A unified MSMS could make it easier for analysts to handle the whole process in one "program" rather than straddling multiple tools. It also enables sharing code for tasks such as cross-validation, which is needed for each of FE, AS, and PT. However, building a unified MSMS poses research challenges that data management researchers are perhaps more familiar with than ML researchers, e.g., the design trade-offs for declarative languages. A unified MSMS also requires ideas from data management and ML, e.g., materializing intermediate data, or sharing computations, which requires RDBMS-style cost-based analyses of ML algorithms. The data management community's expertise in designing query optimizers could be useful here. A caveat is that the three tasks, especially FE, involve a wide variety of operations. It is perhaps infeasible to capture all such operations in the first design of a unified MSMS. Thus, any such MSMS must be extensible. Next, we provide more background on FE, AS, and PT.

[1] Like how IP acts as the "narrow waist" of the Internet.

2. MORE BACKGROUND

Feature Engineering (FE) is the process of converting raw data into a precise feature vector that provides the domain of the prediction function (a learned ML model) [5]. FE includes a variety of *options* (a sequence of computational operations), e.g., counting words or selecting a feature subset. Some options, such as subset selection and feature ranking, are well studied [9]. FE is considered a domain-specific "black art" [4, 5], mostly because it is influenced by many technical and logistical factors, e.g., data and application properties, accuracy, time, interpretability, and company policies. Unfortunately, there is not much integrated systems support for FE, which often forces analysts to write scripts in languages external to data management systems, sample and migrate data, create intermediate data, and track their steps manually [4, 12]. Such manual effort slows and inhibits exploration.

Algorithm Selection (AS) is the process of picking an ML model, i.e., an *inductive bias*, that fixes the hypothesis space of prediction functions explored for a given application [10]. For example, logistic regression and decision trees are popular ML techniques for classification applications. Some ML models have multiple learning algorithms; for example, logistic regression can use both batch and stochastic gradient methods. Like FE, AS depends on both technical and non-technical factors, which leads to a combinatorial explosion of choices. Learning ensembles of ML models is also popular [10].

This complexity often forces analysts to iteratively try multiple ML techniques, which often leads to duplicated effort, and wasted time and resources.

Parameter Tuning (PT) is the process of choosing the values of (hyper-)parameters that many ML models and algorithms have. For example, logistic regression is typically used with a parameter known as the *regularizer*. Such parameters are important because they control accuracy-performance trade-offs, but tuning them is challenging partly because the optimization problems involved are usually non-convex [10]. Thus, analysts typically perform ad hoc manual tuning by iteratively picking a set of values, or by using heuristics such as *grid search* [10]. Some toolkits automate PT for popular ML techniques, which could indeed be useful for some applications. But from our conversations with analysts, we learned that they often tend to avoid such "black box" tuning in order to exercise more control over the accuracy-performance trade-offs.

3. RESEARCH CHALLENGES AND DESIGN TRADE-OFFS

We discuss the key challenges in realizing our vision of a unified MSMS and explain how they lead to novel and interesting research problems in data management. We also outline potential solution approaches, but in order to provide a broader perspective, we explain the design trade-offs involved rather than choosing specific approaches.

3.1 Steering: Declarative Interface

The first major challenge is to make it easier for analysts to specify sets of logically related MSTs using the power of declarative interface languages. There are two components to this challenge.

Language Environment: This component involves deciding whether to create a new language or embedded domain-specific languages (DSLs). The former offers more flexibility, but it might isolate us from popular language environments such as Python, R, and Scala. A related decision is whether to use logic or more general dataflow abstractions as the basis. The latter might be less rigorous but they are more flexible. The lessons of early MSMSs suggest that DSLs and dataflow abstractions are preferable; for example, Columbus provides a DSL for R [18]. The expertise of the data management community with declarative languages will be crucial here.

Scope and Extensibility: Identifying the right declarative primitives is a key challenge. Our goal is to capture a wide spectrum of automation. Thus, we need several predefined primitives to hardcode common operations for each of FE, AS, and PT as well as popular ways of combining MSTs. For example, for FE, standardization of features and joins are common operations, while subset selection is a common way of combining MSTs. For AS and PT, popular combinations and parameter search heuristics can be supported as first-class primitives, but we also need primitives that enable analysts to specify custom combinations based on their expertise. Of course, it is unlikely that one language can capture all ML models for AS or all operations for FE and PT. Moreover, different data types (structured data, text, etc.) need different operations. A pragmatic approach is to start with a set of most common and popular operations as first-class citizens that are optimized, and then expand systematically to include more. Thus, the language needs to be extensible, i.e., support user-defined functions, even if they are less optimizable.

3.2 Execution: Optimization

To fully exploit declarativity, an MSMS should use the relationship between MSTs to optimize the execution of each iteration. Faster iterations might encourage analysts to explore more MSTs, leading to more insights. This challenge has three aspects.

Avoiding Redundancy: Perhaps the most important and interesting aspect is to avoid redundancy in both data movement and computations, since the MSTs grouped together in one iteration are likely to differ only slightly. This idea has been studied before, but its full power is yet to be exploited, especially for arbitrary sets of MSTs. For example, Columbus [18] demonstrated a handful of optimizations for multiple feature sets being grouped together during subset selection over structured data. Extending it to other aspects of FE as well as to AS and PT is an open problem. For example, we need not build a decision tree from scratch for different height parameters, if monotonicity is ensured. Another example is sharing computations across different linear models. Redundancy can also be avoided within a single MST; for example, combining the FE option of joins with the AS option of learning a linear model could avoid costly denormalization [15] or even whole input tables [16]. Extending this to other ML models is an open question. Such optimizations require complex performance trade-offs involving data and system properties, which might be unfamiliar to ML researchers. Thus, the exper-

tise of the data management community in designing cost models and optimizers is crucial here.

System Flexibility: This aspect relates to what lies beneath the declarative language. One approach is to build an MSMS on top of existing data platforms such as Spark, which might make adoption easier, but might make it more daunting to include optimizations that need changes to the system code. An alternative is to build mostly from scratch, which would offer more flexibility but requires more time and software engineering effort. This underscores the importance of optimizations that are generic and easily portable across data platforms.

Incorporating Approximation: Many ML models are robust to perturbations in the data and/or the learning algorithm, e.g., sampling or approximations. While such ideas have been studied for a single MST, new opportunities arise when multiple MSTs are executed together. For example, one could "warm start" models using models learned on a different feature subset [18]. A more challenging question is whether such warm starting is possible across different ML models. Finally, new and intuitive mechanisms to enable analysts to trade off time and accuracy can be studied by asking analysts to provide desired bounds on time or accuracy in the declarative interface. The system could alter its search space on the fly and provide interactive feedback. Exploiting such opportunities requires characterization of new accuracy-performance trade-offs, which might require the data management community to work more closely with ML researchers.

3.3 Consumption: Managing Provenance

Operating on more MSTs per iteration means the analyst needs to consume more results and track the effects of their choices more carefully. But thanks to declarativity, the system can offer more pervasive help for such tasks. This has two aspects.

Capture and Storage: The first aspect is to decide what to capture and how to store it. Storing information about all MSTs naively might cause unacceptable storage and performance overheads. For example, even simple subset selection operations for FE on a dataset with dozens of features might yield millions of MSTs. One approach is to design special provenance schemas based on the semantics of the declarative operations. Another approach is to design new compression techniques. The analyst might have a key role to play in deciding exactly what needs to be tracked; for example, they might not be interested in PT-related changes, but might want to inspect AS- and FE-related changes. Novel applications are possible if these problems are solved, e.g., auto-completion or recommendation of MST changes to help analysts improve Steering. The expertise of the data management community with managing workflow and data provenance could be helpful in tackling such problems. Building applications to improve analyst interaction using provenance might require more collaboration with the human-computer interaction (HCI) community.

Reuse and Replay: Another aspect is the interaction of provenance with optimization. A key application is avoiding redundancy *across* iterations by reusing intermediate data and ML models. Such redundancy can arise, since MSTs typically differ only slightly across iterations, or if there are multiple analysts. This could involve classically hard problems such as relational query equivalence, but also new problems such as defining hierarchies of "subsumption" among ML models. For example, it is easy to reuse intermediate results for logistic regression if the number of iterations is increased by the analyst, but it is non-obvious to decide what to reuse if she drops some data examples or features. This points to the need to characterize a formal notion of "ML provenance," which is different from both data and workflow provenance. The data management community's expertise with formal provenance models could be helpful in tackling this challenge.

Summary. Building a unified MSMS to expose the full power of our abstraction requires tackling challenging research problems, which we outlined with potential solutions and design trade-offs. Our list is not comprehensive – other opportunities also exist, e.g., using visualization techniques to make Steering and Consumption easier. We hope to see more of such interesting new problems in MSMS research.

4. EXISTING LANDSCAPE

We now briefly survey the existing landscape of ML systems and discuss how our vision relates to them. We classify the systems into six categories based on their key goals and functionalities. Due to space constraints, we provide our survey in a separate report [14], but summarize it in Table 1. Our taxonomy is not intended to be exhaustive, but to give a picture of the gaps in the existing landscape.

Numerous systems have focused on efficient and/or scalable implementations of ML algorithms and/or R-like languages. Some other systems have focused

Category	Sub-category	Description	Examples
Packages of ML Implementations	Statistical Software Packages	Software toolkits with a large set of implementations of ML algorithms, typically with visualization support	SAS, R, Matlab, SPSS
	Data Mining Toolkits	Software toolkits with a relatively limited set of ML algorithms, typically over a data platform, possibly with incremental maintenance	Weka, AzureML, ODM, MADlib, Mahout, Hazy-Classify
	Developability-oriented Frameworks	Software frameworks and systems that aim to improve developability, typically from academic research	GraphLab, Bismarck, MLBase
	SRL Frameworks	Implementations of statistical relational learning (SRL)	DeepDive
	Deep Learning Systems	Implementations of deep neural networks	Google Brain, Microsoft Adam
	Bayesian Inference Systems	Systems providing scalable inference for Bayesian ML models	SimSQL, Elementary, Tuffy
Linear Algebra-based Systems	Statistical Software Packages	Systems offering an interactive statistical programming environment	SAS, R, Matlab
	R-based Analytics Systems	Systems that provide R or an R-like language for analytics, typically over a data platform, possibly with incremental maintenance	RIOT, ORE, SystemML, LINVIEW
Model Management Systems		Systems that provide querying, versioning, and deployment support	SAS, LongView, Velox
Systems for Feature Engineering		Systems that provide abstractions to make feature engineering easier	Columbus , DeepDive
Systems for Algorithm Selection		Systems that provide abstractions to make algorithm selection easier	MLBase, AzureML
Systems for Parameter Tuning		Systems that provide abstractions to make parameter tuning easier	SAS, R, MLBase, AzureML

Table 1: Major categories of ML systems surveyed, along with examples from both products and research. It is possible for a system to belong to more than one category, since it could have multiple key goals.

on "model management"-related issues, which involve logistical tasks such as deployment and versioning. A few recent systems aim to tackle one or more of FE, AS, and PT – Columbus, MLBase, DeepDive, and AzureML. However, they either do not abstract the whole process of model selection as we do, or do not aim to support a wide portion of the automation spectrum. We have already discussed Columbus and MLBase in Section 1. DeepDive provides a declarative language to specify factor graph models and aims to make FE easier [17], but it does not address AS and PT. Automation of PT using massive parallelism has also been studied [7]. AzureML provides something similar, and it also aims to make it easier to manage ML workflows for algorithm selection [1]. All these projects provided the inspiration for our vision. We distill their lessons as well as our interactions with analysts into a unifying abstract framework. We also take the logical next step of envisioning a unified MSMS based on our framework to support FE, AS, and PT in an integrated fashion (Figure 1).

5. CONCLUSION

We argue that it is time for the data management community to look beyond just implementing ML algorithms efficiently and help improve the iterative process of model selection, which lies at the heart of using ML for data applications. Our unifying abstraction of model selection triples acts as a basis for designing a new class of analytics systems to manage model selection in a holistic and integrated fashion. By leveraging three key ideas from data management research – declarativity, optimization, and provenance – such model selection management systems could help make model selection easier and faster. This could be a promising direction for interesting and impactful research in data management, as well as its intersection with ML and HCI.

6. REFERENCES

[1] Microsoft Azure ML. `studio.azureml.net`.
[2] Oracle R Enterprise. `www.oracle.com`.
[3] SAS Report on Analytics. `sas.com/reg/wp/corp/23876`.
[4] M. Anderson et al. Brainwash: A Data System for Feature Engineering. In *CIDR*, 2013.
[5] P. Domingos. A Few Useful Things to Know about Machine Learning. *CACM*, 2012.
[6] X. Feng et al. Towards a Unified Architecture for in-RDBMS Analytics. In *SIGMOD*, 2012.
[7] Y. Ganjisaffar et al. Distributed Tuning of Machine Learning Algorithms Using MapReduce Clusters. In *LDMTA*, 2011.
[8] A. Ghoting et al. SystemML: Declarative Machine Learning on MapReduce. In *ICDE*, 2011.
[9] I. Guyon et al. *Feature Extraction: Foundations and Applications*. New York: Springer-Verlag, 2001.
[10] T. Hastie et al. *Elements of Statistical Learning: Data mining, inference, and prediction*. Springer-Verlag, 2001.
[11] J. Hellerstein et al. The MADlib Analytics Library or MAD Skills, the SQL. In *VLDB*, 2012.
[12] S. Kandel et al. Enterprise Data Analysis and Visualization: An Interview Study. *IEEE TVCG*, 2012.
[13] T. Kraska et al. MLbase: A Distributed Machine-learning System. In *CIDR*, 2013.
[14] A. Kumar et al. A Survey of the Existing Landscape of ML Systems. *UW-Madison CS Tech. Rep. TR1827*, 2015.
[15] A. Kumar et al. Learning Generalized Linear Models Over Normalized Data. In *SIGMOD*, 2015.
[16] A. Kumar et al. To Join or Not to Join? Thinking Twice about Joins before Feature Selection. In *SIGMOD*, 2016.
[17] C. Ré et al. Feature Engineering for Knowledge Base Construction. *IEEE Data Engineering Bulletin*, 2014.
[18] C. Zhang et al. Materialization Optimizations for Feature Selection Workloads. In *SIGMOD*, 2014.
[19] Y. Zhang et al. I/O-Efficient Statistical Computing with RIOT. In *ICDE*, 2010.

Participant Privacy in Mobile Crowd Sensing Task Management: A Survey of Methods and Challenges

Layla Pournajaf, Daniel A. Garcia-Ulloa, Li Xiong, Vaidy Sunderam
Math&CS Department
Emory University, Atlanta, GA
{lpourna, dgarci8, lxiong, vss}@emory.edu

ABSTRACT

Mobile crowd sensing enables a broad range of novel applications by leveraging mobile devices and smartphone users worldwide. While this paradigm is immensely useful, it involves the collection of detailed information from sensors and their carriers (i.e. participants) during task management processes including participant recruitment and task distribution. Such information might compromise participant privacy in various regards by identification or disclosure of sensitive attributes – thereby increasing vulnerability and subsequently reducing participation. In this survey, we identify different task management approaches in mobile crowd sensing, and assess the threats to participant privacy when personal information is disclosed. We also outline how privacy mechanisms are utilized in existing sensing applications to protect the participants against these threats. Finally, we discuss continuing challenges facing participant privacy-preserving approaches during task management.

1. INTRODUCTION

The recent increase in the use of smart phones and other mobile devices has created the opportunity to collectively sense and share information for common good. *Mobile crowd sensing* (MCS) refers to the wide variety of sensing models in which individuals with sensing and computing devices are able to collect and contribute valuable data for different applications [30]. MCS is also closely related to location-aware crowdsourcing [38, 2, 48] in which jobs are distributed to workers with regard to their locations. Examples of such applications are crowd-contributed instant news coverage, finding parking spots, monitoring traffic, and crime mapping. In MCS, a *participant* or *carrier* is an individual who collects and contributes data using a sensing device (e.g. a smart phone) that she carries. Collected data is consumed by *end users* directly or after processing by some applications.[1] Mobile crowd sensing can be categorized based on the involvement of participants in sensing actions as *participatory* or *opportunistic*. In a participatory sensing, participants agree to fulfill the requested sensing activities, and are thus explicitly involved in the sensing action (e.g. taking a picture or entering data). In an opportunistic sensing, data is collected by the device with minimum or no involvement of the participants (e.g. reporting speed while driving). Opportunistic sensing could run as a background process, so collecting data requires no interaction with the individuals carrying the sensing devices. From a different point of view, MCS can also be categorized based on the data collection target into *social sensing* and *environmental sensing*. In social sensing applications, a participant collects data about herself (e.g. her vital signs, sport activities) or social interactions (e.g. traffic patterns, parking spots) while in environmental sensing, she monitors certain aspects of the environment (e.g. air pollution, potholes).

To facilitate or coordinate the interaction between applications and participants,[2] a *task management* paradigm is needed to define tasks based on the application requirements, recruit qualified participants, distribute tasks, and possibly coordinate with participants until task completion. One of the major challenges in task management is to ensure a certain degree of privacy for participants. Such an assurance of privacy would increase the disposition of the participants to engage in MCS activity, receive tasks and contribute data, and would ultimately lead to more effective applications.

In this paper, we discuss participant privacy concerns and solutions in the context of task management in mobile crowd sensing. Previous surveys on privacy in participatory sensing applications [17, 16] mainly consider privacy issues related to data col-

[1] In this paper, we use the terms end user and application interchangeably

[2] In this paper we refer to these individuals as participants regardless of the sensing model (participatory or opportunistic)

lection and briefly mention anonymous task distribution, while our main focus is participant privacy during task management. To our knowledge, this is the first survey dedicated to participant privacy issues of task management in MCS. Our main contributions can be summarized as follows:

1. We present a detailed classification of task management in mobile crowd sensing covering all aspects of tasks and distribution methods.
2. We identify the categories of privacy threats to participants of MCS and provide a detailed classification of privacy mechanisms for each type of threat.
3. We discuss ongoing research directions and further challenges in the area of participant privacy in MCS task management.

The rest of this paper is organized as follows. In Section 2 we review and categorize task management models in MCS. We then investigate privacy threats in different tasking schemes in Section 3 followed by existing and applicable privacy solutions studied in Section 4. We discuss limitations of participant privacy in task management and other challenges in Section 5. Finally, Section 6 provides some concluding remarks.

2. TASK MANAGEMENT IN MOBILE CROWD SENSING

We identify the following three entities in task management in mobile crowd sensing:

1. *Participants* are entities that use a sensing device to obtain or measure the required data about a subject of interest.
2. *Applications* or end users are the entities that request data through tasks and then utilize the information acquired by participants.
3. *Tasking entities* are responsible for distribution of tasks to participants who meet the requirements of applications. In certain architectures, end users and participants can also act as tasking entities.

Figure 1 shows the general structure of the task flow in MCS. Task management in crowd sensing can be studied from two perspectives: the type of sensing tasks and the distribution model.

2.1 Sensing Task Schemes

Tasks can be classified into several categories based on features inherent to the tasks or the involved tasking entities. In this study, we classify tasks

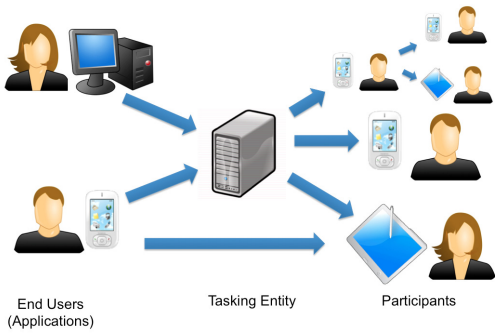


Figure 1: General structure of the task flow in mobile crowd sensing. Note that end users and participants can also act as tasking entities.

along two major dimensions: event based vs continuous, and spatial vs non-spatial. We should note that these dimensions are independent of each other and any combination is possible.

2.1.1 *Event-based vs Continuous*

One way to categorize different tasks is by the frequency with which the data is requested. The frequency could either be event-based or continuous.

Event-based tasks are triggered when a particular situation occurs. This includes special circumstances such as the presence of a participant at a specific location or an ad hoc incident. For example, the tasking entity could ask participants to act as citizen journalists and submit images or other information from a scene of interest when an event occurs [20].

Continuous tasks receive information from the participants periodically or frequently. For example, data could be requested every few minutes to monitor the speed of cars on a specific highway [42], or vital signs of a patient can be frequently requested to track the development of an illness [7]. Continuous tasks have been gaining popularity to keep a record of the different activities performed by the participants. Ganti et.al. have developed a software architecture that keeps track of the participant's activity and location using a personal wearable monitoring system [29], which can have safety, personal, and entertainment applications.

2.1.2 *Spatial vs Non-Spatial tasks*

In location-based tasks, the location of the participant plays an important role in determining task initiation, distribution, or assignment while non-spatial tasks can be triggered by time or other circumstances.

Spatial tasks require the participant to be at a

specific place in order to fulfill a task. With the increasing use of smart phones with integrated GPS, the number of applications in which, tasks are assigned based on the location of participants has also grown. Examples of spatial tasks include those in which sensors such as GPS and accelerometers are positioned in vehicles to detect road conditions. Some of these tasks are opportunistic; they run in the background with little or no involvement from the participant, and they could be used to detect traffic speed, bumps, inclination, and elevation of the road [25, 62, 42]. In contrast, participatory tasks could ask the users to report potholes or the quality of the road as they drive around in their normal commute [26, 72]. Spatial tasks are not restrained to reporting road conditions. For example, a participatory spatial task could require that the participants search for the best prices located at different stores and report them to provide other users with the best prices in the region [21, 9].

Non-spatial tasks are assigned independently of the location of the participant. For example, non-spatial tasks could opportunistically monitor the participant's activities as well as certain aspects of her lifestyle [29]. Tracking a participant's movement and physiological conditions has several beneficial applications in patients with neuromotor disorders [59]. A novel example of opportunistic non-spatial tasks is a "sociometer", which requires the participants to wear sensors that register their face-to-face interactions with other participants [14]. These sensors are able to register parameters such as who the participant is talking to, and how long her conversations last. Afterwards, this information is analyzed to understand the social structure of the participants, and determine how information is diffused, how group problems are solved, and how the community reaches a consensus or forms coalitions.

2.2 Task Distribution Models

Task management models can be categorized according to the way tasks are distributed among participants. The three major categories for these models are: centralized vs decentralized vs hybrid [17], push vs pull, and autonomous vs coordinated.

2.2.1 Centralized vs Decentralized vs Hybrid

In a *centralized model*, a central server or tasking entity provides the participants with different tasks to perform [42]. For example, in a party thermometer application, a central server could choose a set of participants attending an event or party, and request that they rate it. These ratings could serve other users who are considering attending this event [20]. One major issue of a central model in the context of privacy preservation is that the server constitutes a single point of failure for interactions between participants and applications should a security breach occurs. This problem can be alleviated by considering a network infrastructure as a central entity as opposed to having a single server [45].

In a *decentralized model*, each participant can become a tasking entity and decide either to perform a task or pass it forward to other participants who might be better-suited to fulfill the task. This decision would be based on certain attributes of other participants such as location, abilities, or the available hardware in their device. A decentralized recruitment model is proposed in [76] which notifies qualified participants of a forthcoming sensing activity. Some participants selected as recruiting nodes distribute the tasks in certain locations, then in a decentralized manner each participant passes the tasks to whoever matches the task criteria. The advantage in a decentralized model is that there is no single point of failure, so a security breach in a communication does not endanger the privacy of all the users.

A hybrid model includes parts of the centralized and the decentralized models. In this scheme, a central server and a set of participants who act as tasking entities build the task management core [25]. A bubble scheme [60] requires a central server to maintain control of the sensing tasks, which are allocated mostly in a decentralized way. In this model, a task is defined and broadcasted in a particular location of interest by a participant. The task is registered in the server, and other participants who move into the location of interest are signaled by the central server and become bubble carriers. These carriers can broadcast the task and can also fulfill them and report the sensed data to the server.

2.2.2 Push vs Pull model

A different classification for task management models is based on the entity that initiates the task. The initiation model can be *push* or *pull*.

Push model based tasks are initiated by a tasking entity via pushing the tasks on the participants' devices. The platform proposed in [20] uses a push and centralized model where executable binaries of opportunistic tasks are pushed to an optimized set of participants based on predefined criteria. The criteria could depend on several factors such as the location of the participants or the time of the day. An application of this model opportunistically registers GPS and accelerometer data obtained with the participants' mobile phones to determine the

conditions of the road and detect road bumps.

Pull model tasks are queried and downloaded by participants at an arbitrary time or location. A pull based task model can be found in [70], where a set of tasks are stored in a central tasking entity and the participants pull this information and decide which tasks to perform. The decision could be based on different criteria such as preferences, location, or the sensors' capabilities. Nericell [62] represents another pull model example, in which the task of opportunistically detecting road conditions such as potholes, traffic, and noise, depend on the participant's driving route and their smart phone's sensors.

2.2.3 Autonomous vs Coordinated

Tasks can also be categorized based on the allocation scheme that is used to distribute the tasks among the participants. Two approaches that we consider are autonomous task selection and coordinated task assignment [65, 66].

Autonomous task selection is an allocation method in which the participants have access to a set of tasks and they autonomously choose one or more tasks to perform. The participants do not necessarily need to inform the task distributing entity of their decision. While this scheme results in participants sharing fewer attributes with the tasking entities and consequently disclosing less private information, the lack of coordination and global optimization for distributing the tasks can decrease the efficiency with respect to sensing cost or global utility. Another major drawback of autonomous task selection is that it can generate bias in the obtained information. For example, people in urban areas might be more inclined to participate in a sensing task due to the greater presence of sensors or smart phones. This bias would directly affect the variables that are being studied, and will have an effect in the quality of the analysis [1].

Coordinated task assignment aims at improving the quality of the sensed data by optimizing the set of participants recruited to perform tasks. This optimization is based on varied criteria including coverage, quality, sensing costs, and credibility of the sensed data [65, 66]. Reddy et al. [67] proposed a recruitment process based on three stages. The first stage finds those participants that meet the minimum requirements, the second stage aims at maximizing the coverage over an area or time period, and the third stage checks the participants' reputation over coverage and data collection. Once the appropriate set of tasks and participants have been chosen, and the participants have performed the tasks, the task manager might review the participants' progress and evaluate them for future recruitment.

3. PRIVACY THREATS IN TASK MANAGEMENT

In mobile crowd sensing, privacy concerns might discourage participants from data contribution. Such concerns include a) disclosure of participant identity, b) disclosure of sensitive attributes such as race, age, or locations (e.g. current location, home or work address), and c) disclosure of more complex information such as personal activities or conditions (e.g. lifestyle or sicknesses). From a different perspective, participant privacy concerns can be aggravated either i) directly via sharing real IDs, IP addresses, exact locations, or other sensitive attributes, or ii) indirectly by sharing insensitive information (e.g. home address inference from trajectories of participants [54]). Designing a task management model that preserves the privacy of participants can be challenging due to the nature of crowd sensing tasks and task distribution models. In this section, we investigate the information that a participant shares with other tasking entities during the task management process and discuss how this information can directly or indirectly breach her privacy. We also discuss the applicability of the privacy threats with respect to the different tasking schemes we discussed in the previous section. Table 1 provides a summary of the privacy threats for vulnerable tasking schemes.

Adversary Models

From the perspective of participant privacy, the adversaries in MCS task management may include some or all of end-users (applications), tasking entities, and other participants based on their involvement in task management. Regardless of the role of adversarial entities, they are generally modeled as either *semi-honest* or *malicious*. Here, we study these two models and privacy threats associated with each. We also discuss how different entities in different task management frameworks fall into these categories.

3.1 Semi-honest Entities

The semi-honest entities (also known as passive) are assumed to follow the task management protocols and would not actively alter the data to breach the privacy of the participants. However, these entities may attempt to exploit any acquired information from participants to learn their private data. We categorize the privacy attacks conducted by semi-

honest entities in task management into *task tracing attacks* and *location-based attacks*, both of which are described below.

3.1.1 Task Tracing Attacks

When a participant downloads specific tasks from a tasking server (i.e. pull-based tasks), shares her preferences during a coordinated task assignment, or notifies a server of accepting a pushed task, she may reveal some attributes such as location, time, the task types in which she is interested, or some attributes of the sensing device she is carrying. For example, if a task is designed for undergraduate students majoring in History and can only be handled by Android devices, performing such a task reveals some information about the participant. This information alone might not breach her privacy; however, linking multiple tasking actions might allow an adversary to trace the selected tasks by the participant and consequently reveal her identity or other sensitive attributes [70]. Continuing the previous example, if the same history student later performs another task for an application designed for Hispanic students at her university, the adversary might be able to infer her identity. Some of the attributes that can be used to trace participants are real names, pseudonyms, International Mobile Equipment Identity (IMEI), IP addresses, or other user/device identifiers. An example of task tracing attack is illustrated in Figure 2.

Some tasking models distribute tasks among the participants based on their behavior and their profile as opposed to a device ID with specific characteristics [39]. Assuming that mobile phones are almost exclusively used only by its owner, the use of the device reflects the user's preferences. In particular, mobility can reflect the user's interest and can be used to determine if the user is more capable of fulfilling a task. However, this tasking model is still prone to location-based attacks.

3.1.2 Location-based Attacks

Spatial tasks requested (i.e. a pulled task) or accepted (i.e. a pushed task) by participants might lead to disclosure of their current location and eventually their sensitive locations such as home/work addresses or even their identification through location-based attacks. Location-based attacks are widely recognized in the context of location-based services (LBS), however, certain attributes of mobile crowd sensing make it more vulnerable to some of spatial attacks. Here, we give a brief review of such attacks in MCS.

In frequent spatial tasks, even if the participant

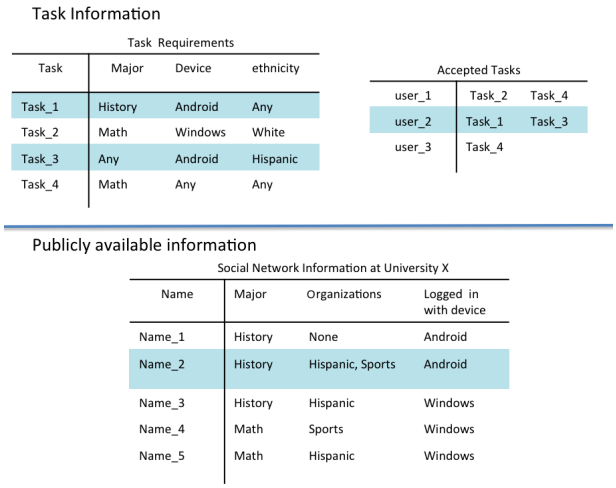


Figure 2: A task tracing attack in MCS task management using user-ids. Accepting task_1 is not enough to determine the identity of user_2, however, tracing the tasks she has performed and using available information from other sources could provide the necessary means to determine her identity.

is using the application anonymously (e.g. using pseudonyms), her trajectory might reveal her sensitive locations or commutes [55] or eventually disclose her identity using location-based de-anonymization attacks [28]. Krumm proposed several algorithms to identify a small group of anonymous participants and the home address of a larger group through location-based inference attacks [54]. They used the distribution of location traces during time, the last destinations of the day and the distribution of stay times to infer the home addresses of the participants. A location could be simply considered as a home if it is visited frequently by the same user at night [12]. Participant locations can also be exploited using trajectory data mining algorithms [61] to identify their significant locations. The trajectory data can be also used to infer the individuals' life patterns (i.e. private schedules or lifestyles) [81].

Continuous or frequent spatial tasks make MCS more prone to location-based inference attacks as more location traces of participants are collected. A simple example of this attack in mobile crowd sensing task management is illustrated in Figure 3.

Kazemi et. al. [46, 47] defined a location-based attack in campaign-based Participatory Sensing applications when participants used Spatial k-anonymity [74] to hide their location. The location attack is defined as the identification of a participant by an untrusted server by learning the location of her issued task query. They observed that all participants of a campaign query spatial tasks from the server (a.k.a. all-inclusivity property) asking for tasks closer to them than any other participants

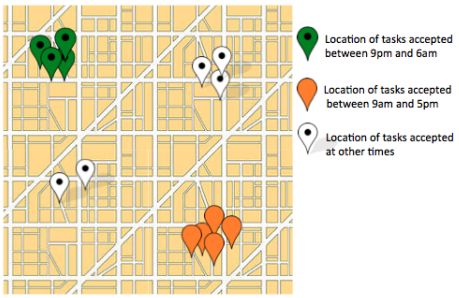


Figure 3: A simple location-based inference attack in MCS task management. The time and location of the accepted tasks can be enough to determine the participant's home and work addresses.

(a.k.a range dependency property). These properties result in the server having spatially-dependent requests from all participants, so they argued that participatory sensing is more vulnerable to such location-based attack. Gonzalez et. al. showed that people and their movements are highly correlated [36] making such attacks possible.

Another location-based attack targets applications that utilize the density distribution of participants (i.e. aggregated number of participants) in a location for task management [71]. This attack exploited by a group of terrorists can be used to identify dense areas for explosive launches.

3.2 Malicious entities

Malicious entities actively try to breach the privacy of participants. Privacy attacks associated with malicious task management entities include both aforementioned attacks along with several active de-anonymization attacks such as malicious tasking and collusion attacks. To prevent or stop these attacks, privacy countermeasures should be plugged into sensing devices or other trusted-parties.

3.2.1 Malicious Tasking

In the process of task definition, a malicious entity might create tasks that impose strict limitations on participant attributes or the device she is carrying (e.g. requiring a special lifestyle or a rare sensor type to qualify for the task). This attack which is called narrow tasking [70] might result in disclosure of identity or other sensitive attributes of the participant who accepts such a specialized task. In another variation of malicious tasking (a.k.a. selective tasking [70]), the tasking entity may share tasks with a limited set of participants to be able to learn their attributes or trace them (e.g. pushing or assigning a task to only one participant).

3.2.2 Collusion Attack

Several applications (end users) or tasking entities might collude to link the information of the participants for de-anonymization of the individuals or acquire their other private information. These attacks known as collusion attacks might be hard to detect in mobile crowd sensing systems since individuals might contribute to different applications using separate task management systems with no control over how their information is shared with others. For example, individuals might share some information with application A1 and other information with application A2 considering none of this information being personally identifiable separately. However, in reality, if applications A1 and A2 share pieces of her information, they might be able to de-anonymize her identity. Moreover, a malicious entity might create several applications with different contexts in an attempt to collect more private data from individuals. To avoid such attacks, while individuals might not be able to stop the collusion, they can at least control the amount of information they share with each application and also the overall information they share with all of the applications. We discuss this concept in detail in Section 4.3.

4. PRIVACY COUNTERMEASURES IN TASK MANAGEMENT

We categorize privacy solutions in MCS task management based on the applicable state-of-the-art privacy mechanisms. These mechanisms can be adopted in MCS based on privacy threats relative to task schemes and distribution models and the privacy preferences of the participants. In other words, there is no privacy-preserving method suitable for every user and application. For example, a participant who uses her real name to register to MCS applications cannot benefit from anonymization techniques. Table 1 summarizes privacy countermeasures that can be used for different privacy threats.

4.1 Anonymization

Anonymization techniques remove or hide identification information from all the interactions between the participant and other entities during task management. We review some of the anonymization techniques here.

4.1.1 Pseudonyms

One of the basic methods to preserve the anonymity of the participants includes using pseudonyms by replacing the identification information with an alias [17]. While this technique prevents location-based inference attacks, it does not protect the par-

Table 1: Summary of privacy threats and countermeasures for different tasking scenarios.

Privacy Threats	Tasking Scenarios	Countermeasures
Task tracing	Pulling specific tasks Coordinated task assignment Push-based tasks with notification	Anonymization Temporally constrained sharing Policy-based privacy preferences
Location-based attacks	Spatial tasks	Spatial cloaking Temporally constrained sharing Private information retrieval Differential privacy Policy-based privacy preferences
Narrow tasking	All tasking schemes	K-Anonymization Policy-based privacy preferences
Selective tasking	Coordinated task assignment Push-based tasks	K-Anonymization Policy-based privacy preferences
Collusion attacks	All tasking schemes	Policy-based privacy preferences

ticipants from task tracing or location-de-anonymization attacks (see Section 3.1.2). For a detailed review of these methods in MCS refer to a recent survey [16].

4.1.2 Connection Anonymization

These methods can be used to avoid tracing attacks using network-based identifiers such as IP addresses or device identifier such as International Mobile Equipment Identity (IMEI), and SIM card identifiers (IMSI, ICC-ID). One such technique which is adopted in crowd sensing applications [70] is onion routing [23] which hides the IP addresses of the participants from the other entities.

4.1.3 K-Anonymization

K-anonymization [73] is an established anonymization technique in database privacy [5]. A user is considered to be k-anonymous if her identity is indistinguishable from $k-1$ other users. In MCS task management, participants can adopt this method to avoid malicious tasking attacks by accepting a task only if there exists $k-1$ other qualifying participants for it. For example, if a user learns that she is the only qualified participant for a task, she would avoid performing it. K-anonymization is also widely adopted for location privacy which is discussed separately in Section 4.2.

4.2 Spatio-Temporal Privacy Methods

With the growing advance of location-based services, several spatio-temporal privacy mechanisms have been developed recently (see recent surveys in [31, 56, 4]). Although the context in mobile crowd sensing is different from location-based services, these mechanisms can be used to address location privacy problems in such scenarios as well. However, since location and time are two crucial pieces of information in an effective task management model, applying the existing spatio-temporal privacy-preserving techniques can be challenging. Here, we study some of the applicable methods in MCS task management.

4.2.1 Spatial Cloaking

In some crowd sensing applications, a perturbed or cloaked location can be used for spatial task management instead of exact locations. Spatial cloaking or perturbation hides the participant location inside a cloaked region using spatial transformations [50], generalization (e.g. k-anonymity) [44, 78], or a set of dummy locations [51] in order to achieve location privacy. Some MCS applications do not require exact locations (e.g. pollution or weather monitoring), but for the majority of the applications with utility depending on location accuracy, adopting cloaking methods remains a challenge. In recent work [65], participants of a coordinated spatial task assignment would share their cloaked location to obtain a set of closest tasks. They developed probabilistic methods to deal with uncertainty for a globally optimized task assignment.

Kazemi et. al. [46, 47] showed that spatial k-anoymity methods used in location-based services are not directly applicable to Participatory Sensing. Therefore, they proposed that a group of the representative participants ask for spatial tasks from an untrusted server, and share their results with the rest of participants. They would also adjust the spatial regions in queries to make queries independent from the location of other participants. Vu et. al. [77] proposed a spatial cloaking mechanism for Participatory Sensing based on k-anonymity and locality-sensitive hashing (LSH) to preserve both locality and k-anonymity.

While most traditional location cloaking methods rely on syntactic privacy models and are subjective to inference attacks, recent works applied more rigorous privacy notion based on differential privacy. The work in [3] proposed a location perturbation method based on a rigorous notion of indis-

tinguishability, which is similar to the differential privacy concept. Another recent work [79] protects the exact locations with differential privacy in a proposed *delta*-location set, which is derived in Markov model to denote the possible locations where a user might appear at any time.

4.2.2 Temporally Constrained Sharing

Some approaches share exact locations for tasking; however, they avoid or mitigate the location based attacks to some extent by controlling the timing of disclosures. For example, to avoid frequent revealing of location of participants in spatial tasks, Krause et al. [52] use a spatial obfuscation approach. In their solution, they divide the space into a set of regions, then with a certain probability distribution, a subset of participants is selected in each region to report their exact location. Such methods can be used in traffic monitoring applications.

Another method [52] assigns spatial tasks to participants in a way that the number of tasks for each participant is minimized. In such an approach, there will be longer intervals between each location disclosure, mitigating location-based inference attacks. This scheme can be further controlled by participants by setting explicit policies regarding the intervals in which they prefer to share their location. We discuss these methods in Section 4.3.

4.2.3 Aggregated Location with Differential Privacy

Differential Privacy [24] is a promising privacy-preserving approach with a strong protection guarantee. This method is adopted in privacy-preserving publishing of statistical information about location-based datasets [31] guaranteeing that individual location information disclosure does not occur. It can also prevent privacy attacks on the aggregated number of participants in a location as discussed in 3.1.2. In recent work, differential privacy is adopted for spatial crowdsourcing task assignment [75] in which a trusted aggregator (e.g. a cell service provider) computes differentially private aggregated counts of participants in various spatial regions and provides them to tasking entities for task assignment.

4.2.4 Private Information Retrieval

In autonomous pull-based tasking schemes, participants can retrieve the best suited tasks without providing their attributes using private information retrieval (PIR). PIR-based methods have been adopted for location-based services recently [31] since they guarantee cryptographic privacy by allowing data retrieval from a database without revealing any information to the database server about the retrieved item. Such an anonymous tasking scheme suffers from overlapping task selection and bias since sharing entities do not learn which tasks are retrieved.

4.3 Policy-based Privacy Preferences

To avoid direct or inference-based privacy breaches, participants should be able to set fine-grained preferences to control information sharing in a way that a curious party cannot learn or infer any private attributes. Such policies may include settings to ignore location-based tasks when the participant is within a specified range of a sensitive location (e.g. home or work), ignore narrow tasks, limit the number of tasks per time periods, or avoid sharing information that could be linked to previously disclosed data.

Shilton et. al. [69] introduced the concept of participatory privacy regulation in MCS which promotes participants' involvement in developing their own privacy policies and setting their personal boundries. Some methods provide a trusted cloud-based storage and processing entity for each participant to store and fully control sharing of her personal information with applications and end users [13, 63, 10]. A recent incentive-based task assignment approach allows participants to set their preferred privacy levels, which are then incorporated into a tasking cost model to limit the frequency of location disclosures (i.e. a task that requires location disclosure will be more costly for a participant with strict privacy preferences) [68].

5. DISCUSSION

In this section, we discuss further research directions and challenges of participant privacy in MCS including the limitations of privacy preserving tasking solutions, and privacy issues related to other components of MCS such as data collection.

5.1 Private Tasking Limitations

5.1.1 Trust and Credibility

Privacy and trust generally follow conflicting goals since the participant's trust is gained by higher accuracy and exactness of provided data, but privacy aims at hiding or perturbing identifying data (which includes majority of exchanged data in MCS) to protect the participant [1, 35]. Furthermore, trust issues become more challenging for anonymous tasking since they may result in tasking to untrustworthy or unqualified participants [17].

A trustworthy privacy-aware framework is proposed in [49], which defines the relationship between

trust and privacy in participatory sensing as a reverse k-nearest problem. Participants' privacy is procured in [34] by installing trusted software on the mobile device to encrypt the data before it is sent to the remote server. While this approach ensures the integrity of data during transmission to the server, the credibility of the participants is not evaluated.

Assessing the reputation of the participants while maintaining their anonymity and preserving their privacy is particularly difficult when a task requires the users to be at a certain location to collect the data more efficiently. Anonymous participants are prone to provide falsified or faulty data and it would be challenging to evaluate their participation, especially if different task actions cannot be linked due to privacy mechanisms [41, 18]. One approach to avoid trust issues in coordinated task management might be to assign a task to several participants to avoid the effect of malicious or faulty participation, however such method would result in a waste of resources.

Huang et. al. [41] proposed an anonymous reputation system for participatory sensing, which preserves the privacy of participants by separating their reputation from their identity. Another recent work [18] also addresses the issue of maintaining the reputation of the anonymous participants by using pseudonyms and anonymous transfer of the reputation information. They also use simulations to analyze the trade-off between privacy and reputation.

5.1.2 Reward-based Tasking

The challenge for rewarding participants in the presence of privacy mechanisms is very similar to the trust challenges since both require participant evaluations. However, trust models need to trace and review participants progress while incentives can be handled per task completion without linking it to other tasks. Several recent privacy-preserving incentive models are proposed in the literature [82, 58].

An anonymous credential system (or pseudonym system) can preserve the privacy of users while allowing internet transactions with service providers [11], so that an incentive-based system that supports privacy can be implemented. Zhang et. al [82] proposed a model based on pseudonym, encryption, and hashing to protect participant privacy.

A delayed rewarding model is proposed in [70] which aims at preventing task-reward linkage. Assuming that only the application can calculate the rewards for each task, their reward scheme includes a payment service that receives an anonymous claim message from the user after one or several tasks have been completed. The anonymity of the message is ensured by the application in the user's device, that encrypts a new identity for the user each time a message is sent. The payment service uses a one-way function to verify the message and forwards the reward to the user. The user's privacy will be preserved if the message is new for each report and the one-way function is secure.

5.1.3 Utility and Efficiency

Privacy mechanisms that obfuscate location, time, or other attributes challenge task management with uncertain or incomplete information. Therefore, the tasking entities may need to task a larger set of participants or conduct more computation to reach a certainty similar to non-private models. A recent work [65, 66] proposed a two-stage tasking model in which participants would share their cloaked locations rather than exact locations. Their model consists of a central tasking server which deals with location uncertainty and recommends globally optimized tasks to participants, and then each participant locally refines and further self-assigns tasks strictly following the global recommendation. Although this model achieves a comparable utility as the non-private method, the sensing and computational costs are higher due to uncertainty.

5.2 Data-related Privacy Concerns

In addition to how tasks are managed, task context (i.e. captured sensor data) might also lead to privacy issues for participants. For instance, noise monitoring tasks might record participants' voices, or if participants continuously report their driving habits during a trip, the destination of the trip may still be inferred even without sharing specific locations [22]. Another example of data-related privacy problems is contributing images that contain identifying information about the participants such as faces or locations [6] which can be de-identified before uploading to protect their privacy [64]. Fine-grained privacy preferences can also help participants to ignore tasks requiring sensitive contexts. Other privacy-preserving data collection solutions such as differential privacy can be used to perturb aggregated data before submitting to a server [80, 27]. If a trusted aggregator is not available, participants can use secure multi-party computation protocols [37] to aggregate their data before submitting to the data collector.

Furthermore, in most applications, captured sensor data contains meta-data such as time/location of individual sensing actions which might result in

location-based inference attacks. By linking reports to participants, other tracing attacks would arise. To assure participant privacy in mobile crowd sensing, privacy-preserving methods should be developed during both task management and data collection. Privacy issues during reporting has been extensively studied in literature, and several privacy-preserving data collection and aggregation methods have been proposed [17, 16].

Hu et. al. proposed a decentralized model to protect the privacy of participants in a social network while reporting data to an untrusted server. In their approach, participants pass data to their friends in a chain-like fashion before it is uploaded to the server. An spatial cloaking method based on generalization is used in [70] to hide the location of participants during data reporting. Huang et. al. [40] argued that location generalization methods decrease the utility of collected data significantly, particularly in traffic data monitoring applications. They proposed an alternative approach based on microaggregation and also a hybrid approach including both generalization and microaggregation.

5.3 Privacy Mechanism Enforcement

In Section 4 we discussed how suitable privacy mechanisms could be determined by the types of threats, but enforcing these mechanisms still remains as a challenge. In MCS, privacy mechanisms can be enforced on sensing devices (participants), semi-honest or trusted tasking entities, or other trusted third-parties. On the other hand, privacy-preserving architecture could be centralized or decentralized [30]. However, different models might introduce further complications and security issues which need to be considered in choosing an enforcement model.

A trusted third-party is one of the commonly used privacy-preserving approaches in MCS. Many works use a centralized server to anonymize the participants information, cloak their locations or perturb the aggregated number of participants in a region [75] while satisfying the users privacy requirements. In these architectures, the tasking entity (entities) receives anonymized information from the trusted party including perturbed or cloaked locations. Other methods use a decentralized architecture in which either participants trust each other and use peer-to-peer methods for spatial cloaking [46, 15] or they benefit from secure multi-party computation [30]. Moreover, a decentralized model may include a group of trusted agents [53] who share a complex data structure [32] to store and enforce privacy policies.

Krontiris et. al. [53] proposed trusted decentralized cloud-based agents to cloak the location of participants and enforce their preferred privacy policies. The agents are organized in a quadtree structure which is stored and managed in a decentralized fashion. While decentralized approaches avoid bottlenecks of centralized methods such as having a single point of failure and scalability problems, they introduce more complications for privacy enforcement and maintenance.

5.4 Privacy-Awareness

Another important topic regarding participants' privacy in crowd sensing task management is the users' privacy awareness. Several studies [54, 43, 19] show that individuals are generally unaware of threats of using location-based services and place a low value on the privacy of their location data. In general, with no or little incentives the participants might willingly share their sensitive location information and moving patterns. Other studies [8, 56] explore participants' attitude in sharing their location for incentives (i.e. the value of location) and their willingness and preferences for using location obfuscation methods for sharing highly sensitive data such as their home or workplace address. Their attitude towards sharing their location data depends on several factors such as the usefulness of the application, the amount of data to be shared, the incentives to share it, and if it will be used for commercial or other purposes [56, 54, 33, 57].

6. CONCLUSION

Mobile crowd sensing is an emerging topic with a wide variety of possible applications. However, the functionality of MCS relies on the participation of individuals who might be concerned about their privacy. In particular, task management as a central part of crowd sensing structure poses several threats to participant privacy, which should be identified and addressed. In this survey, we have classified different potential privacy risks and outlined their solutions for task management in MCS in an effort to raise awareness and preserve the privacy of the participants.

Acknowledgment

This research is supported by the Air Force Office of Scientific Research (AFOSR) DDDAS program under grant FA9550-12-1-0240.

7. REFERENCES

[1] C. C. Aggarwal and T. Abdelzaher. Social sensing. In *Managing and Mining Sensor Data*, pages 237–297. Springer, 2013.
[2] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz. Location-based crowdsourcing: extending

crowdsourcing to the real world. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, pages 13–22. ACM, 2010.
[3] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914. ACM, 2013.
[4] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. Di Vimercati, and P. Samarati. Location privacy protection through obfuscation-based techniques. In *Data and Applications Security XXI*, pages 47–60. Springer, 2007.
[5] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 217–228. IEEE, 2005.
[6] A. Besmer and H. Richter Lipford. Moving beyond untagging: photo privacy in a tagged world. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1563–1572. ACM, 2010.
[7] L. Brown, B. Grundlehner, J. van de Molengraft, J. Penders, and B. Gyselinckx. Body area network for monitoring autonomic nervous system responses. In *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on*, pages 1–3. IEEE, 2009.
[8] A. Brush, J. Krumm, and J. Scott. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 95–104. ACM, 2010.
[9] N. Bulusu, C. T. Chou, S. Kanhere, Y. Dong, S. Sehgal, D. Sullivan, and L. Blazeski. Participatory sensing in commerce: Using mobile camera phones to track market price dispersion. In *Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense 2008)*, pages 6–10, 2008.
[10] R. Cáceres, L. Cox, H. Lim, A. Shakimov, and A. Varshavsky. Virtual individual servers as privacy-preserving proxies for mobile devices. In *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, pages 37–42, 2009.
[11] J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 21–30, 2002.
[12] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from gps data. *Proceedings of the VLDB Endowment*, 3(1-2):1009–1020, 2010.
[13] H. Choi, S. Chakraborty, Z. M. Charbiwala, and M. B. Srivastava. Sensorsafe: a framework for privacy-preserving management of personal sensory information. In *Secure Data Management*, pages 85–100. Springer, 2011.
[14] T. Choudhury and A. Pentland. Sensing and modeling human networks using the sociometer. In *2012 16th International Symposium on Wearable Computers*, pages 216–216. IEEE, 2003.
[15] C.-Y. Chow, M. F. Mokbel, and X. Liu. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. *GeoInformatica*, 15(2):351–380, 2011.
[16] D. Christin. Privacy in mobile participatory sensing: Current trends and future challenges. *Journal of Systems and Software*, 2015.
[17] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, 84(11):1928–1946, 2011.
[18] D. Christin, C. Roßkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere. Incognisense: An anonymity-preserving reputation framework for participatory sensing applications. *Pervasive and mobile Computing*, 9(3):353–371, 2013.
[19] G. Danezis, S. Lewis, and R. J. Anderson. How much is location privacy worth? In *WEIS*, volume 5. Citeseer, 2005.
[20] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. Prism: Platform for remote sensing using smartphones. pages 63–76. in Proceedings of the 8th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), 2010.
[21] L. Deng and L. P. Cox. Livecompare: grocery bargain hunting through participatory sensing. In *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, page 4. ACM, 2009.
[22] R. Dewri, P. Annadata, W. Eltarjaman, and R. Thurimella. Inferring trip destinations from driving habits data. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 267–272. ACM, 2013.
[23] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.
[24] C. Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.
[25] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN)*, 6(1):6, 2009.
[26] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 29–39. ACM, 2008.
[27] L. Fan and L. Xiong. Real-time aggregate monitoring with differential privacy. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2169–2173, 2012.
[28] S. Gambs, M.-O. Killijian, and M. Núñez del Prado Cortez. De-anonymization attack on geolocated data. *Journal of Computer and System Sciences*, 2014.
[29] R. K. Ganti, P. Jayachandran, T. F. Abdelzaher, and J. A. Stankovic. Satire: a software architecture for smart attire. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 110–123. ACM, 2006.
[30] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. *Communications Magazine, IEEE*, 49(11):32–39, 2011.
[31] G. Ghinita. *Privacy for Location-Based Services*. Synthesis Lectures on Information Security, Privacy, and Tru. Morgan & Claypool, 2013.
[32] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Mobihide: a mobilea peer-to-peer system for anonymous location-based queries. In *Advances in Spatial and Temporal Databases*, pages 221–238. Springer, 2007.
[33] A. Ghosh and A. Roth. Selling privacy at auction. *Games and Economic Behavior*, 2013.
[34] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall. Toward trustworthy mobile sensing. In *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, pages 31–36. ACM, 2010.
[35] P. Gilbert, J. Jung, K. Lee, H. Qin, D. Sharkey, A. Sheth, and L. P. Cox. Youprove: authenticity and fidelity in mobile sensing. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 176–189. ACM, 2011.
[36] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.
[37] S. Goryczka and L. Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 2015.
[38] J. Howe. *Crowdsourcing: How the power of the crowd is driving the future of business*. Random House, 2008.
[39] W.-J. Hsu, D. Dutta, and C. Ahmed Helmy. A paradigm for behavior-oriented profile-cast services in mobile networks. *Ad Hoc Networks*, 10(8):1586–1602, 2012.
[40] K. L. Huang, S. S. Kanhere, and W. Hu. Towards privacy-sensitive participatory sensing. In *Pervasive Computing and Communications, 2009. IEEE International Conference on*, pages 1–6. IEEE, 2009.
[41] K. L. Huang, S. S. Kanhere, and W. Hu. A privacy-preserving reputation system for participatory sensing. In *Local Computer Networks, 2012 IEEE 37th Conference on*, pages 10–18, 2012.
[42] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko,

A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: a distributed mobile sensor computing system. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 125–138. ACM, 2006.
[43] E. Kaasinen. User needs for location-aware mobile services. *Personal and ubiquitous computing*, 7(1):70–79, 2003.
[44] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *Knowledge and Data Engineering, IEEE Transactions on*, 19(12):1719–1733, 2007.
[45] A. Kansal, M. Goraczko, and F. Zhao. Building a sensor network of mobile phones. pages 547–548. in Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN), 2007.
[46] L. Kazemi and C. Shahabi. A privacy-aware framework for participatory sensing. *ACM SIGKDD Explorations Newsletter*, 13(1):43–51, 2011.
[47] L. Kazemi and C. Shahabi. Towards preserving privacy in participatory sensing. In *Pervasive Computing and Communications Workshops, 2011 IEEE International Conference on*, pages 328–331, 2011.
[48] L. Kazemi and C. Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 189–198. ACM, 2012.
[49] L. Kazemi and C. Shahabi. Tapas: Trustworthy privacy-aware participatory sensing. *Knowledge and information systems*, 37(1):105–128, 2013.
[50] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *Advances in Spatial and Temporal Databases*, pages 239–257. Springer, 2007.
[51] H. Kido, Y. Yanagisawa, , and T. Satoh. An anonymous communication technique using dummies for location-based services. In Proceedings of the IEEE International Conference on Pervasive Services, 2005.
[52] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 481–492. IEEE, 2008.
[53] I. Krontiris and T. Dimitriou. Privacy-respecting discovery of data providers in crowd-sensing applications. In *Distributed Computing in Sensor Systems, 2013 IEEE International Conference on*, pages 249–257, 2013.
[54] J. Krumm. Inference attacks on location tracks. In *Pervasive Computing*, pages 127–143. Springer, 2007.
[55] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.
[56] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.
[57] C. Li, D. Y. Li, G. Miklau, and D. Suciu. A theory of pricing private data. *ACM Transactions on Database Systems (TODS)*, 39(4):34, 2014.
[58] Q. Li and G. Cao. Providing privacy-aware incentives for mobile sensing. In *Pervasive Computing and Communications, 2013 IEEE International Conference on*, pages 76–84, 2013.
[59] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, M. Welsh, et al. Mercury: a wearable sensor network platform for high-fidelity motion analysis. In *SenSys*, volume 9, pages 183–196, 2009.
[60] H. Lu, N. D. Lane, S. B. Eisenman, and A. T. Campbell. Bubble-sensing: Binding sensing tasks to the physical world. *Pervasive and Mobile Computing*, 6(1):58–71, 2010.
[61] H. J. Miller and J. Han. *Geographic data mining and knowledge discovery*. CRC Press, 2009.
[62] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 323–336. ACM, 2008.
[63] M. Mun, S. Hao, N. Mishra, K. Shilton, J. Burke, D. Estrin, M. Hansen, and R. Govindan. Personal data vaults: a locus of control for personal data streams. In *Proceedings of the 6th International Conference*, page 17. ACM, 2010.
[64] E. M. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *Knowledge and Data Engineering, IEEE Transactions on*, 17(2):232–243, 2005.
[65] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka. Spatial task assignment for crowd sensing with cloaked locations. In *Proceedings of the 2014 International Conference on Mobile Data Management. IEEE*, 2014.
[66] L. Pournajaf, L. Xiong, V. Sunderam, and X. Xu. Stac: Spatial task assignment for crowd sensing with cloaked participant locations. In *23rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2015.
[67] S. Reddy, D. Estrin, and M. Srivastava. Recruitment framework for participatory sensing data collections. pages 138–155. in Proceedings of the 8th International Conference on Pervasive Computing, Springer Berlin Heidelberg, May 2010.
[68] M. Riahi, T. G. Papaioannou, I. Trummer, and K. Aberer. Utility-driven data acquisition in participatory sensing. EDBT/ICDT, ACM, March 2013.
[69] K. Shilton, J. A. Burke, D. Estrin, M. Hansen, and M. Srivastava. Participatory privacy in urban sensing. 2008.
[70] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos. Anonysense: A system for anonymous opportunistic sensing. *Journal of Pervasive and Mobile Computing*, 7(1):16–30, 2010.
[71] R. Shokri, G. Theodorakopoulos, J. Le Boudec, and J. Hubaux. Quantifying location privacy. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 247–262. IEEE, 2011.
[72] G. Strazdins, A. Mednis, G. Kanonirs, R. Zviedris, and L. Selavo. Towards vehicular sensor networks with android smartphones for road surface monitoring. In *2nd International Workshop on Networks of Cooperating Objects (CONET11), Electronic Proceedings of CPS Week*, volume 11, 2011.
[73] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):571–588, 2002.
[74] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
[75] H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *Proceedings of the VLDB Endowment*, 7(10), 2014.
[76] G. S. Tuncay, G. Benincasa, and A. Helmy. Autonomous and distributed recruitment and data collection framework for opportunistic sensing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 16(4):50–53, 2013.
[77] K. Vu, R. Zheng, and L. Gao. Efficient algorithms for k-anonymous location privacy in participatory sensing. In *INFOCOM, 2012 Proceedings IEEE*, pages 2399–2407, 2012.
[78] S. Wang and X. S. Wang. In-device spatial cloaking for mobile user privacy assisted by the cloud. In *Mobile Data Management, 2010 Eleventh International Conference on*, pages 381–386. IEEE, 2010.
[79] Y. Xiao and L. Xiong. Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1298–1309, 2015.
[80] L. Xiong, V. Sunderam, L. Fan, S. Goryczka, and L. Pournajaf. Predict: Privacy and security enhancing dynamic information collection and monitoring. *Procedia Computer Science*, 18:1979–1988, 2013.
[81] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. In *Mobile Data Management: Systems, Services and Middleware, 2009. Tenth International Conference on*, pages 1–10. IEEE, 2009.
[82] J. Zhang, J. Ma, W. Wang, and Y. Liu. A novel privacy protection scheme for participatory sensing with incentives. In *Cloud Computing and Intelligent Systems, 2012 IEEE 2nd International Conference on*, volume 3, pages 1017–1021, 2012.

Cleanix: a Parallel Big Data Cleaning System

Hongzhi Wang
Harbin Institute of Technology
wangzh@hit.edu.cn

Mingda Li
Harbin Institute of Technology
limingda@hit.edu.cn

Yingyi Bu
University of California, Irvine
yingyib@ics.uci.edu

Jianzhong Li
Harbin Institute of Technology
lijzh@hit.edu.cn

Hong Gao
Harbin Institute of Technology
honggao@hit.edu.cn

Jiacheng Zhang
Tsinghua University
chinahitzjc@gmail.com

ABSTRACT

For big data, data quality problem is more serious. Big data cleaning system requires scalability and the ability of handling mixed errors. Motivated by this, we develop Cleanix, a prototype system for cleaning relational Big Data. Cleanix takes data integrated from multiple data sources and cleans them on a shared-nothing machine cluster. The backend system is built on-top-of an extensible and flexible data-parallel substrate— the Hyracks framework. Cleanix supports various data cleaning tasks such as abnormal value detection and correction, incomplete data filling, de-duplication, and conflict resolution. In this paper, we show the organization, data cleaning algorithms as well as the design of Cleanix.

1. INTRODUCTION

Recent popular Big Data analytics applications are motivating both industry and academia to design and implement highly scalable data management tools. However, the value of data not only depends on the quantity but also relies on the quality. On one side, due to the high volume and variation, those Big Data applications suffer more data quality issues than traditional applications. On the other side, efficiently cleaning a huge amount of data in a shared-nothing architecture has not been well studied yet. Therefore, to improve the data quality is an important yet challenging task.

Many data cleaning tools [1] have been proposed to help users to detect and repair errors in data. Although these systems could clean data effectively for many datasets, they are not suitable for cleaning Big Data due to the following three reasons. First, none of the existing systems can scale out to thousands of machines in a shared-nothing manner. Second, various error types such as incompleteness, inconsistency, duplication, and value conflicting may co-exist in the Big Data while most existing systems are ad-hoc. As examples, CerFix [2] focuses on inconsistency while AJAX [3] is for de-duplication and conflict resolution. The last but not least, existing systems often requires users to have specific data cleaning expertise. For example, CerFix [2] requires users to understand the concept of conditional functional dependency (CFD), while AJAX [3] needs users to express data cleaning tasks with a declarative language. However, many real-world users do not have a solid data cleaning background nor understand the semantics of a specific data cleaning language.

In order to address the fundamental issues in existing systems and support data cleaning at a large scale, we design and implement a new system called Cleanix. The key features of Cleanix are listed as follows.

- **Scalability**. Cleanix performs data quality reporting tasks and data cleaning tasks in parallel on a shared-nothing cluster. The backend system is built on-top-of Hyracks [4], an extensible, flexible, scalable and general-purpose data parallel execution engine, with our user-defined data cleaning second-order operators and first-order functions.
- **Unification**. Cleanix unifies various automated data repairing tasks for errors by integrating them into a single parallel dataflow. New cleaning functionalities for newly discovered data quality issues could be easily added to the Cleanix dataflow as either user-defined second-order operators or first-order functions.
- **Usability**. Cleanix does not require users to be data cleaning experts. It provides a simple and friendly graphical user interface for users to select rules with intuitive meanings and high-level descriptions. Cleanix also provides a bunch of visualization utilities for users to better understand error statistics, easily locate the errors and fix them.

The main goal of this demonstration is to present the Cleanix system architecture and execution process by performing a series of data integration and cleaning tasks. We show how the data cleaning operators are used to clean data integrated from multiple data sources.

2. RELATED WORK

Big data has been very popular among academical and industrial fields. However, due to the features of

four *V*s, which means Volume, Velocity, Variety and Value, we may face the problems of data quality easily and need to detect and solve the errors in data. The detection of error means to find dirty items. According to the difference among methods, there are three types of methods.

- **Entity identification** The entity identification means to find the different items representing the same thing in the real world. By entity identification, we can detect the phenomenon of duplication. There has already been several methods for entity identification [5].
- **Error detection according to rules** To utilize rules during the detection we can use variable kinds of rules such as the functional dependencies [6], conditional functional dependencies [7], and so on. [7] designs an auto-detection algorithm based on the SQL language to find the items going against the conditional functional dependencies and extending inclusion dependencies.
- **Error detection based on master data** The main data is a high-quality data set to provide a synchronous consistent view. For example, [8] gives a relatively complete theory to describe the integrity of main data and the complexity of the users' queries.

The repairing for error means the modification or supplement to the data with error to improve the quality. According to different thoughts, the methods for repairing can be divided into three parts.

- **Repairing by rules** Repairing by rules mainly means to modify the data and make it satisfy the rules provided by managers. The [9] provides a repairing algorithm GREEDY_REPAIR. The algorithm is expended from the above method and uses the conditional functional dependencies for repairing. The [10] repairs the inconsistent data by the graph theory.
- **Truth Discovery** To solve the conflict data during entity resolution, we use truth discovery algorithm. The [11] uses the iteration method to calculate the truth degree of the source and the self-confidence degree of the value. [12] considers dependency among data sources, which is calculated from the self-confidence of the value.
- **Machine learning** Machine learning methods are mainly used for repairing incomplete data. The methods based on machine learning include decision tree, Bayesian network and Neural Network.

[13] is the demo plan of this paper. This paper introduces the detail of design and techniques in Cleanix.

3. SYSTEM OVERVIEW

3.1 Data Cleaning Tasks

Cleanix aims to handle four types of data quality issues in a unified way:

- *Abnormal value detection and correcting* is to find the anomalies according to users' options of rules and modify them to a near value that coincides with the rules.
- *Incomplete data filling* is to find the empty attributes in the data and fill them with proper values.
- *De-duplication* is to merge and remove duplicated data.
- *Conflict resolution* is to find conflicting attributes in the tuples referring to the same real-world entity and find the true values for these attributes. For example, tuples referring to the same person may have different values in age, but only one value should be chosen.

We believe that these four data cleaning tasks cover most data quality issues. Note that even though some data errors could not be processed directly such as nonconcurrency and inconsistency, one can take care of them by dynamically deploying new first-order user-defined functions into our system. For example, nonconcurrency can be processed as conflict resolutions among the data referring to the same real-world entity.

3.2 The Hyracks Execution Engine

We use Hyracks as backend to accomplish the above tasks efficiently at large scales. Hyracks is a data-parallel execution engine for Big Data computations on shared-nothing commodity machine clusters. Compared to MapReduce, Hyracks has following advantages:

- **Extensibility**. It allows users to add data processing operators and connectors, and orchestrate them into whatever DAGs. However, in the MapReduce world, we need to cast the data cleaning semantics into a scan (map)—group-by(reduce) framework.
- **Flexibility**. Hyracks supports a variety of materialization policies for repartitioning connectors, while MapReduce only has local file system blocking-materialization policy and HDFS materialization policy. This allows Hyracks to be elastic to different cluster configurations., e.g., behaving like a parallel database style optimistic engine for small clusters (e.g., 200 machines) but a MapReduce style pessimistic engine for large clusters (e.g., 2000 machines).
- **Efficiency**. The extensibility and flexibility together lead to significant efficiency potentials., e.g., one can implement the hybrid-hash style conflict resolution on Hyracks but not on MapReduce.

Several cloud computing vendors are developing non-MapReduce parallel SQL engines such as Impala [1] and Stinger [2] to support fast Big Data analytics. However, these systems are like "onions" [14]—one cannot directly use their internal Hyracks-like engines under the SQL skin for data cleaning. However, the Hyracks software

[1] https://github.com/cloudera/impala
[2] http://hortonworks.com/blog/100x-faster-hive/

stack is like a layered "parfait" [14] and Cleanix is yet-another parfait layer on-top-of the core Hyracks layer.

3.3 Cleanix Architecture

Cleanix provides web interfaces for users to input the information of data sources, parameters and rule selections. Data from multiple data sources are preprocessed and loaded into a distributed file system—HDFS[3]. Then each slave machine reads part of data to start cleaning. The data cleaning dataflow containing second-order operators and connectors is executed on slaves according to the user specified parameters and rules (e.g., first-order functions). At end of dataflow, the cleaned data are written to HDFS. Finally, cleaned data are extracted from HDFS and loaded into desired target database.

4. THE SYSTEM INTERNALS

In this section, we discuss the details of the Cleanix data cleaning tasks, pipeline, the algorithmic operators and the profiling mechanism.

4.1 Integration of Data

Before cleaning data, we need to download data to our system. We can support downloading from different kinds of databases including MySQL and MSSQL. While integrating items from different source databases, there will be a problem of conflicts among primary keys. To solve this, we add a new column as new primary key and a new column to show source of an item.

4.2 Data Cleaning Algorithms

The four data cleaning tasks including abnormal value detection and correcting, incomplete data filling, de-duplication and conflict resolution all have their own implementation algorithms. These algorithms are developed for parallel platform. We will introduce solutions to the four tasks respectively as following.

4.2.1 Abnormal Value Detection and Correcting

Before executing the part, users can select data type for each attribute from types provided by Cleanix. Meanwhile, they can set cleaning rules for attributes. The rules include the detection rule for data type, legal range and the correcting rule after detecting abnormal data. According to the rules, we can detect and do simple filling in this cleaning part. For example, to the Numeric(integer or float) Attribute: *Detection Rule*: Set the maximum number and minimum number to detect abnormal value. *Filling Rule*: **a**. Simple Filling: Fill with fixed number or date. **b**. Intelligent Filling: According to attributes provided by users, which are relevant to the abnormal attribute, we find similar items. Using the items' attributes relevant to the abnormal value, we get suitable result by choosing the biggest, smallest, most frequent, least frequent, average number of them.

[3] `http://en.wikipedia.org/wiki/Apache_Hadoop`

4.2.2 Data Imputation Algorithm

To fill incomplete data, the system needs to find items similar to the incomplete items. For big data, we need an efficient algorithm to obtain the similarity. There are two common algorithms called Edit Distance and 2-Gram Table. We choose the 2-Gram Table [15] and give up the Edit Distance due to its high complexity ($O(n^2)$), working process not suitable for parallelization(compare one string with all others) and drawback for some common phenomenons such as reversal of names. To build the 2-Gram table, we firstly build an empty Hash Table with two columns called Key and Number. Each item of Key Column is a String and each item of the Number Column's is an Integer Array. Then, for all the items, we read strings and take each string's 2-character substring as Hash's key. If there has already been such a key in the table, we add the number to the end of corresponding array. Otherwise, we add a new item into the table with the key and number of the string.

With the 2-Gram table, the similarity between strings can be calculated by the following process: **Firstly**, build another Hash table. There are two columns in the table called Key and Amount. Each item of Key Column is an integer to show the number of a string. Each item of Amount Column an integer to record the number of the same substrings. **Then**, traverse all 2-character substrings. Regard substring *s* as Key and inquire it in the 2-Gram table we have got. **Thirdly**, if there is such a Key in 2-Gram table, we regard the items in the corresponding array as key and insert them into Hash table. We initialize the corresponding amount as 1. If there is already such key, we add the corresponding amount with 1. **Finally**, after traverse to strings, we traverse the Hash table we build this time. Key represents string and Amount represents the number of the same substrings. So the bigger Amount represents more similar string.

After getting the similarities, if it is larger than a threshold, we regard the item similar to the incomplete item. Put all similar items into an ArrayList. According to filling rule set by users, we complete the imputation.

4.2.3 De-duplication Algorithm

De-duplication is also called entity identification. We divide this duty into two parts. One part is the Grouping Algorithm and the other part is the Merging In A Group.

Grouping Algorithm The kernel idea is to group the whole data. We put the similar items into the same group. Meanwhile, we promise different items are in different groups. So the First Problem is to separate the items which are similar but represent different things.

The Second Problem is to put the items, which are not similar but represent the same thing, together. For the First Problem, we firstly use the Grouping Algorithm to put them together. For the Second Problem, we set a formula to calculate the similarity. We add other attributes and let users to set the weight. Thus, we can ensure the items, which are not similar but represent the same thing, in one group. In our algorithm, users can set many attributes as relevant attributes in De-duplication. By setting weight for each relevant attribute and a threshold, if the similarity calculated is larger than the threshold, we think the two items are the same and put them in the same group.

Merging In One Group After grouping, master node will get an array whose elements represent a set for similar items.Then we do the Merging In One Group. **Firstly**, Master Node traverses the sets in the array generated from the Grouping Algorithm. **Then**, Master Node use the greedy method to send each set to different Slave Node and try to make each node deal the same amount of data. **Finally**, each Slave Node uses the effective clustering approach introduced in [16]. We can know the complexity of the Merging In One Group algorithm is $O(n)$. The main theory is that if there are two items' similar substrings are more than a threshold in a group, we will regard them as duplicate items.

4.2.4 Conflict Resolution Algorithm

Conflict Resolution is to solve the conflicts while merging many duplicate items to one item. During the process, some attribute may be different.The system will automatically choose the strings which appear most frequently. Meanwhile, the users can also set their own rules to solve the problem. For example, to date attribute, users can choose the earliest, latest, most frequent and least frequent date to fill in. The working process of the Conflict Resolution is as following: **Firstly**, Master Node sends the users' rules to each slave node. **Then**, Slave Nodes traverse the repeated items set. If there is any conflict among the items, we solve the conflicts by the rules set by users. If there is no user rule, we can automatically choose the most frequent one. **Finally**, each Slave Node sends back the items to Master Node and Master Node collects the items from Slave Nodes and get the final result.

4.2.5 Share Information and Output Result

Share information among cluster In the working process for parallel cluster, we always need to share information. The easiest method shown in Figure 1 is to let all slave nodes send their own information to the master node. Then, the master node sends the integrated information back. This method is easy, but has some problems. For example, when many slave nodes send data to master node in the meantime, there may be blocking for the network and the overlap of memory in master node. Therefore, we design an annular transmission algorithm to solve the above problem as the Figure 2. We suppose there are m nodes called NC_1, NC_2....NC_m. Suppose the data they store is n_1,....n_m. We only need to circle for m-1 times to finish sharing information. In each circling, each Slave Node only needs to send his own data.

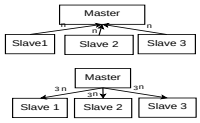


Figure 1: Primary Method to Send Data

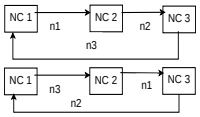


Figure 2: Improved Method to Send Data

Output Result During the Conflict Resolution, we have finished the data collection. But when we collect items from different sources, we will face a problem of conflict to the Key. So we build a new ID column as a new key to solve the conflict. Meanwhile, we will add the column describing the data source.

4.3 Data Processing Ordering

To make the discussion brief, we use A, I, D and C to represent the modules of the process of abnormal value detection and correcting, incomplete data filling, de-duplication and conflict resolution, respectively. The order of four tasks of data cleaning in Cleanix is determined with the consideration of effectiveness and efficiency. These four modules could be divided into two groups. Module A and I are in the same group (Group 1) sharing the same detection phase since the detection of abnormal values and empty attributes can be accomplished in a single scan of the data. Module D and C are in the same group (Group 2) since the identifications of entities with the entity resolution operator are required for both de-duplication and conflict resolution. De-duplication merges tuples with the same entity identification while conflict resolution is to find true values for conflicting attributes for the different tuples referring the same entity identification. The reason why Group 1 is executed before Group 2 is that the repairation of abnormal values and empty attributed will increase the accuracy of entity resolution. In Group 1, Module A is before I since abnormal values interfere the incomplete attribute filing and lead to incorrect fillings. In Group 2, Module D is before C since only when different tuples referring to the same entity are found and grouped, the true values of conflicting attributes could be found.

4.4 Dataflow

The dataflow graph is shown in Figure 3. The dataflow has 8 algorithmic operators and 4 stages, where the computation of each stage is "local" to each single

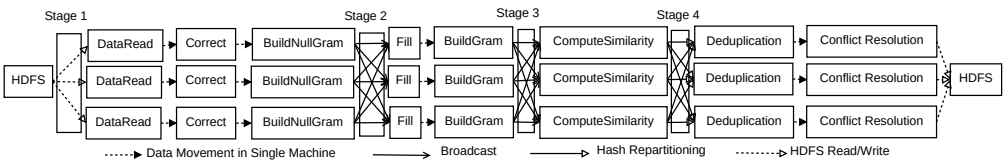


Figure 3: The Cleanix Dataflow Graph

machine and the data exchange (e.g., broadcast or hash repartitioning) happens at the stage boundaries. In following part, we illustrate the algorithmic operators and the rules for each stage in topological order in Figure 3.
Stage 1.This is performed on each slave machine.

- *DataRead*. It scans incoming file splits from the HDFS. The data are parsed and translated into the Cleanix internal format.
- *Correct*. This is blocking operator—data are checked according to the rules selected by users to detect the abnormal values and incomplete tuples. When an abnormal value is detected, it is corrected according to corresponding revision rules (first-order functions). When an incomplete tuple is encountered, it is identified for further processing.
- *BuildNullGram*. This operator builds an inverted list for all incomplete tuples for the imputation based on similar tuples. The inverted list is called *gram table*. It is a hash table in which the k-gram is the key and the id set of tuples containing such a k-gram is the value.

Stage 2. The incoming broadcast connector to this stage broadcasts the gram tables such that all slaves share the same global gram table.

- *Fill*. For each tuple with incomplete attribute, similar tuples are found according to the gram table. The incomplete attribute is filled with the aggregated value of the corresponding attribute in similar tuples according to the imputation rules (first-order functions) selected by users such as average, max or the most frequent.
- *BuildGram*. A local gram table is built for the local data for the attributes potentially containing duplications or conflicts, which are chosen by users. Since a local gram table has been built with BuildNullGram operator, only the newly filled values of corresponding attributes are scanned in this step.

Stage 3.The local gram tables are broadcast to make all slaves share the same global gram table. Note: only updated values in local gram tables are broadcast in Stage3.

- *ComputeSimilarity*. The similarities between each local tuple and other tuples are computed according to the global gram table. When the similarity between two tuples is larger than a threshold, they are added to the same group and form many groups finally.

Stage 4.Groups are partitioned according to hashing value of bloom filter of the union of gram sets in group.

- *De-duplication*. A weighted graph G is built to describe the similarity between tuples in each group. Similar vertices are merged iteratively in G until no pairs of vertices can be merged [16]. This step is executed iteratively until the ratio between the number of shared connected vertices and the number of the adjacent vertices of each vertex is smaller than a threshold. The tuples corresponding to all merged vertices are considered as duplications.
- *Conflict Resolution*. Tuples corresponding to the merged vertices are merged. During merging, when an attribute with conflicting values is detected, it is resolved with voting according to selected rules chosen by users. Options (first-order functions) include max, min, average and the most frequent.

4.5 Data Cleaning Result

In the final part of the demonstration, we illustrate the exploration of data cleaning results and interaction of user and the system. More specifically, Cleanix will compare the repaired data with the original ones. The original and modified data are distinguished in different colors. When the user selects a modified value, the modifications are shown. Additionally, the user could modify the data. The modifications are merged when the cleaned data is transmitted from HDFS to the target database.

Besides, users can also check the data quality in high level. We can see how the violations are distributed among the data by different histograms and statistical categorization for both attribute and tuple level.

5. INTERFACE

The system allows users to add several machines to clean data from different data sources. You can input the name of new Slave Nodes in the NodeController Name as shown in Figure 4. And if we are cleaning data stored in different machine, we can input the IP address of the machine, Port, Username and Password to get access to the data at the bottom of the same page shown in Figure 4. After setting the database connection information, we can set the cleaning rules to find abnormal value and do filling as Figure 5 shows. Users can also set the weight and threshold to do de-duplication like Figure 6.

After setting the basic information for data cleaning, we can click the button *See the status of the working system and you need to start it to work here* and open the page like Figure 7. We can start working there and find the status of system. When system finishes cleaning, we can check the data cleaning result by inputting the range of item's ID. Cleaning results include four types of dirty items including *abnormal value*, *duplication*, *incomplete value* and *conflict*. Part of the four types of cleaning results is shown in Figure 8.

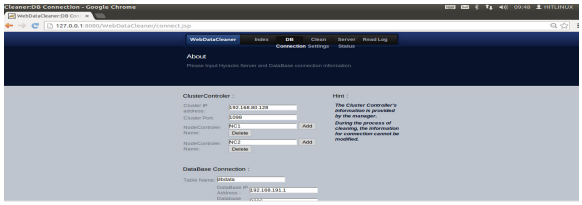


Figure 4: Add New Slave Nodes and Data Source

Figure 5: Set Data Cleaning Rules

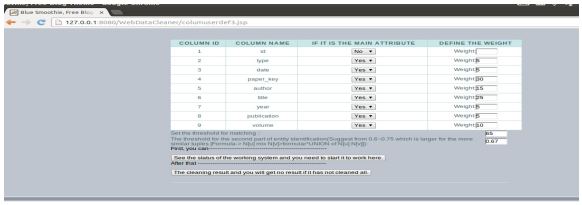


Figure 6: De-duplication

Figure 7: See the Working Status of the System

Acknowledgements. This paper was supported by NGFR 973 grant 2012CB316200 and NSFC grant 61472099,61133002.

Figure 8: *Part of Cleaning result* : Incomplete Value and Conflict

6. REFERENCES

[1] Thomas N. Herzog, Fritz J. Scheuren, and William E. Winkler. *Data quality and record linkage techniques*. Springer, 2007.

[2] Wenfei Fan, Jianzhong Li, Shuai Ma, Nan Tang, and Wenyuan Yu. CerFix: A system for cleaning data with certain fixes. *PVLDB*, 4(12):1375–1378, 2011.

[3] Helena Galhardas, Daniela Florescu, Dennis Shasha, Eric Simon, and Cristian-Augustin Saita. Declarative data cleaning: Language, model, and algorithms. In *VLDB*, pages 371–380, 2001.

[4] Vinayak R. Borkar, Michael J. Carey, Raman Grover, Nicola Onose, and Rares Vernica. Hyracks: A flexible and extensible foundation for data-intensive computing. In *ICDE*, pages 1151–1162, 2011.

[5] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[6] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[7] Philip Bohannon, Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. Conditional functional dependencies for data cleaning. In *ICDE*, pages 746–755, 2007.

[8] Wenfei Fan and Floris Geerts. Relative information completeness. *ACM Trans. Database Syst.*, 35(4):27, 2010.

[9] Philip Bohannon, Michael Flaster, Wenfei Fan, and Rajeev Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005*, pages 143–154, 2005.

[10] Gao Cong, Wenfei Fan, Floris Geerts, Xibei Jia, and Shuai Ma. Improving data quality: Consistency and accuracy. In *Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria, September 23-27, 2007*, pages 315–326, 2007.

[11] Amélie Marian and Minji Wu. Corroborating information from web sources. *IEEE Data Eng. Bull.*, 34(3):11–17, 2011.

[12] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. Integrating conflicting data: The role of source dependence. *PVLDB*, 2(1):550–561, 2009.

[13] Hongzhi Wang, Mingda Li, Yingyi Bu, Jianzhong Li, Hong Gao, and Jiacheng Zhang. Cleanix: A big data cleaning parfait. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 2024–2026, 2014.

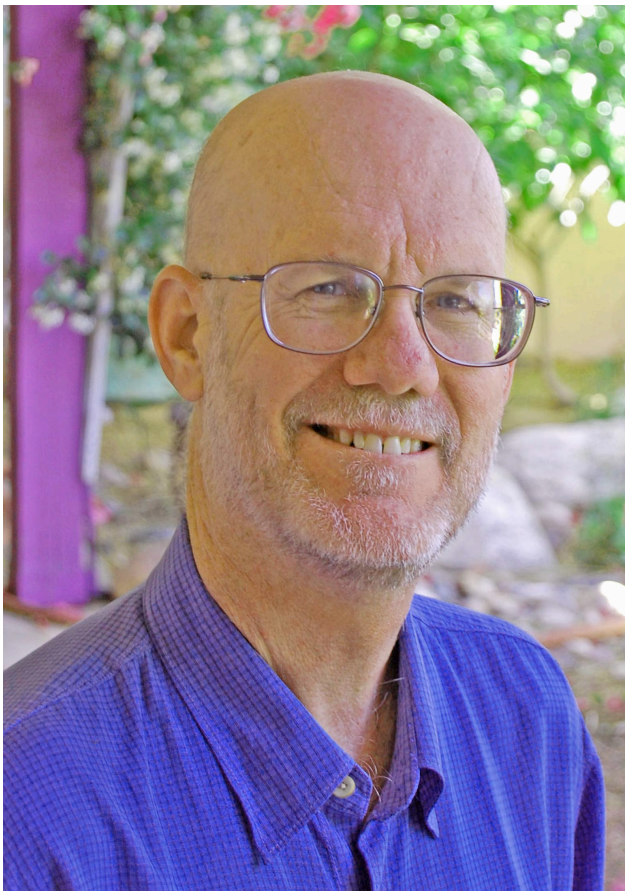
[14] Vinayak R. Borkar, Michael J. Carey, and Chen Li. Inside "Big Data management": ogres, onions, or parfaits? In *EDBT*, pages 3–14, 2012.

[15] Esko Ukkonen. Approximate string matching with q-grams and maximal matches. *Theor. Comput. Sci.*, 92(1):191–211, 1992.

[16] Lingli Li, Hongzhi Wang, Hong Gao, and Jianzhong Li. EIF: A framework of effective entity identification. In *WAIM*, pages 717–728, 2010.

Rick Snodgrass Speaks Out on Standards, Personal Brands and Science

Marianne Winslett and Vanessa Braganholo



Rick Snodgrass
http://www.cs.arizona.edu/people/rts/

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we're in Phoenix, site of the 2012 SIGMOD and PODS conferences. I have here with me Rick Snodgrass, who is a professor of computer science at the University of Arizona. Rick has served as the Editor-in-Chief of ACM Transactions on Database Systems, the chair of ACM SIGMOD, the ACM Pubs Board and the ACM History Committee. He has received the SIGMOD Outstanding Contributions Award and ACM Outstanding Contribution Award and he's an ACM Fellow. Rick's PhD is from Carnegie-Mellon University.

So, Rick, welcome!

Thank you!

Rick, you are best known for your work on temporal databases, and you even worked hard to get temporal constructs into the SQL standard. What role do you think database researchers should play in the standards community?

I think it's very important for the database researchers to have a role. Unfortunately, by the way that the Standards Committee is set up (it's run and funded by vendors), it's very hard to spend time with them. They're open to having people come in, but it has to be funded by the researchers. So I would actually like to have the committee be more accepting of people from our research community and actually pay for them to come in and invite them in. For example, the ANSI standards are closed. They don't even release them to the community for comment until they have been finalized. I think there should be much more of a dialog with the research community.

There is really no excuse for not having an accurate and complete provenance on the ideas that you're working on.

Are there "mistakes" that we could have prevented if we would have been more involved with that?

The standard is a very big standard. It is thousands of pages long. I think perhaps had we been involved, we could have found more foundational aspects to reduce the redundancy in the standards. So that would have been one place. I don't know if that is a mistake, but certainly the more input is better, I think.

A standard that is thousands of pages long? It sounds like almost contradicting terms to me. How can anyone understand?

I don't think anyone totally understands. I think it's about four thousand pages. I think the standards body (the people that did it), who have been working on it for decades understand it very well. Us normal mortals, I don't think so.

Isn't there a gap between what the standard says and what people actually implemented?

Yes, so it's much bigger than what most DBMSs implement. Although each DBMS implements a portion of it, of course there are a lot of inconsistencies between the standard and the implementations.

So in what sense is it a standard if people don't implement all of it and then the parts they do implement are inconsistent?

Well it's better than having a free-for-all where they are all different. So there is some compatibility. So we should be happy for that.

So I guess in industry there are incentives for diversity. That people can't switch between products so easily. What are the incentives for standardization?

The standards process is very interesting because you have all these vendors -- who are competing with each other -- getting in the same room and trying to come up with something that is a standard. That really shouldn't work. So it's amazing how well it does work, given that situation. They are all very much competitors and they each want to win. So the fact that we get anything at all, I think we should celebrate.

People are saying that the relational temporal database constructs don't apply to graph databases (for example, in biology). Do you agree with that?

No, I don't. Actually I've done some work with temporal XML graph data. I think the underlying concepts, like sequenced, apply directly. Well, databases have foreign keys and that's a lot like a tree structure also, or graph structure. So I think that if we get to the foundational concepts, hopefully it should apply, with modification, anywhere.

What parts need to be changed to move to the graph databases?

I think the ideas need to be applied to that specific place. So when we did our work with temporal XML, we had to figure out what were the unique aspects of XML, but there were actually very few. If you have an XQuery query on XML, you can say "I want to evaluate that on a temporal document at each point in time". That is what *sequenced* does. So that applies directly. Now, how you actually implement that efficiently is a complex problem, but at least you know what the semantics should be.

You have a project right now on ergalics. What is that?

Ergalics is from the Greek word "erg" which means work or tool. And so it's taking computational tools like DBMSs or compilers, or other kinds of tools. It's a science of computational tools. The reason we needed this word is because computer science has the word "science". So the science of computer science is a very awkward formulation. So I came up with this word, "ergalics". I did a Google search and no one had used the word, so it's a new word.

Computer science really has three different perspectives in it. One is mathematics… so dependency theory, asymptotic complexity, etc. Another one is engineering. Most of what we do is engineering. Engineering is doing something better, faster, cheaper. So you're always trying to do a better job. Most of PODS papers, for instance, are in the mathematical perspective; most of SIGMOD is in the engineering. We have actually very little science methodology or predictive theories. That's where I think we need to also go. That does not mean that the other two perspectives aren't just as valid. Mathematics can oftentimes inform science and science can inform engineering. Right now that centerpiece is not there. So we jump from mathematics to engineering. I think that really good engineers have intuitive understanding of the predictive rules by which these computational tools work, but we haven't yet articulated those.

So what would they look like in the case of databases?

So I've been working on that for quite a while. It's very difficult research in that whenever I find something I want see "why is that?" and then I look in the code. The way I like to think about this is to treat a DBMS like a biologist would treat a rabbit. So a biologist looks at a rabbit and says, "this rabbit has big ears". Why might that be? Well maybe it is because they need to hear better. Well, but there are other animals that need to hear better that don't have big ears. So why do rabbits? Especially in the desert? Desert rabbits have the biggest ears. So maybe it's because they need to release heat. Well, then you can do studies and experiments, in that case, blood flow in the ears could make a difference. A biologist does not say, "How do I make a rabbit run faster? Maybe if I make the legs longer…". That's not a question you ask. You ask "why". Why is the rabbit the way it is? So I'm looking for predicative theories about DBMSs that would apply across DBMSs. So, SQLServer, DB2, Oracle, Teradata…these are implemented by different people. They are totally different products. What are things we can say about them in general?

Okay, but rabbits are evolving, but slowly. These DBMSs are evolving super fast; at least we hope they're evolving fast…

Actually most of their foundational parts have been around for close to 30 years. I mean, cost-based query optimization, locking, concurrency control… So actually, there is a lot that has remained pretty steady. And so we can ask questions about them. For instance, if you add a query operator, a relational operator, what will happen to the number of optimal plans?

You mean a new kind of operator? That's like adding a fifth leg to a rabbit.

This is informing the engineering. So right now we would say, "We want to make some queries run faster". So I'll add a certain kind of index or I'll add another kind of relational operator. That has the benefit of making some queries run faster. So that's good. It also makes the optimizer more complex. So we're going to have more mistakes: sub-optimal plans. So do you get more advantage or disadvantage? Is there a point in which when you are adding an operator, on average you actually slow down the DBMS? That's a question that you really can't ask about a single DBMS, you have to ask it across the class.

And what's the answer to that one?

I'm still working on that one. So we don't know that. That's a very interesting question for engineering. That's just one of many.

Well at four thousand pages, maybe we're approaching the point where the next thousand pages would just bring it downhill.

Yes, as a matter of fact one of the complaints against *ergalics* is that well, these are programs. We understand programs. I mean we can look and see exactly how they work. Well a DBMS is so complex, no one understands it. Science deals with the universe, which is also very complex. They have a whole bunch of methodologies that we can use to understand these very complicated things.

So can you give me an example of an insight that you've gotten to date, using this approach?

It looks like as you add operators, you do have an increase in sub-optimality. So it looks like there is a limit at which point, we get diminishing returns and it actually goes the other way.

Do you know yet whether we've already reached the limit?

I haven't gotten the theory to the point where it's that specific yet.

In the database research community we don't often think about branding, but I know that you do. So I guess I'll start with a little story. Jiawei Han told me that we needed to change the same of our research group at Illinois and we did. We used to be called the Database and Information Systems Group, and now we're the Data and Information Systems Group. Jiawei said that if we called ourselves database group, everyone who is not in this area thinks that databases are a solved problem and therefore we're not really doing anything interesting. So this may be an example of a branding issue. So what do you think we need to do in the database community in terms of our brand?

So I think brands are very important. Since we're in the southwest we first need to define what branding is. It's not what you do to cattle to identify them. We're not talking about burning signs on their hides. We're talking about identifying in the minds of people what does this discipline or what does this person do. My wife is a marketing professor so that's how I know about branding. Disciplines have brands, departments have brands, universities have brands, and people have brands. And I think it is very important for people to think about what their brand is. As far as a discipline, I totally agree with Jiawei that DBMSs are viewed as a solved problem. We're still trying to do better, we're still trying to do more engineering, but DBMSs are wonderfully efficient and powerful tools right now. So going from DBMSs or databases to data I think is exactly the right place where our discipline needs to go. We really understand data very well and I think that that is a skill that the world needs, scientists need for their data, engineers need it for their data, and even experimental mathematicians need it for their data. We have a lot to give the whole world.

So we'd be the Data Management Research or Information Management Research?

So I see information as being data with insight. You're adding insight. There is a hierarchy from data to information to knowledge to wisdom. I think we are experts at the first couple of levels. You get in the philosophy when you get up to the wisdom part or morality or whatever, but certainly those first few steps are very important. I think we have a lot to add.

So do we need to change our current brand or is it already in the right place?

Well this is the Special Interest Group on Management of Data. It's not Management of Databases. So yes, SIGMOD I think has the right name. As opposed to for instance IEEE Data Engineering. Well, that's kind of restrictive. They're only looking at a third of the thing, whereas we can look at the mathematics of data, the science of data and the engineering of data.

Well all those people out there who would say, "I'm in the database group". What should they be saying instead?

I mean if they're interested in databases and doing that, that's fine! I don't think that us as a society, but also as a general discipline should necessarily limit ourselves that way.

So a new name is called for?

Or just emphasizing MOD "Management of Data" and going back to that.

> ***A biologist does not say, "How do I make a rabbit run faster? […]". That's not a question you ask. You ask "why". Why is the rabbit the way it is?***

What about personal branding as researchers? How should we handle that?

So I wrote a paper[1] with my wife, Merrie Brucks, on this, with a whole bunch of different approaches for personal branding. The goal here is to come up with a single phrase that when people hear that phrase, they think of you. And when they think of you, they think of that phrase. So, in my career, I've branded myself as "temporal databases". So when people need a temporal database guy on their program committee, they think of me and they think of some other people. And when they think of me, they think of temporal databases. How have I helped that? I've written glossaries, I've written surveys, I've written papers, most of which have the words "temporal database" if not in the title, then in the abstract. I'm now going towards "science of computer science". That's what I've been working on the last few years. That's an awkward phrase, so I

[1] Richard T. Snodgrass and Merrie L. Brucks, "Branding Yourself," *ACM SIGMOD Record* 33(2):117–125, June 2004.

came up with a new word. If you Google "ergalics" you'll see my name. And eventually when you think of Rick Snodgrass, maybe you'll think of ergalics. A lot of people have a problem with this because they see it as limiting, but you get to pick what you want to be associated with you.

Well then let's pick on Jim Gray because he is not here. So what is Jim Gray's brand? He's arguably our most successful researcher.

I would argue his brand is "integration of research approaches". He talked to everyone. He knew pretty much what everyone was doing and he could help them figure out how they fit into the big picture and he could articulate that big picture. So that's what I see his brand is. So a brand can be a topic, it can be an approach, it can be a methodology, and it can be a special ability like Jim Gray's…

Okay. Can we pick on some other people who are maybe a little more here? So what's David Dewitt's brand?

David Dewitt is very articulate and controversial. When he says something, people want to hear what he says. And he's brilliant so he's really good at bringing problems to the community and bringing solutions, and bringing places where we are not doing very well, which is very helpful for us. He's a fantastic person to be on a panel for this reason.

I think one can go too far and just spending all of one's time reading and one can go too far and not read anything. It requires some real skill to figure out where that middle ground is.

Oh yeah, he's exciting to listen to.

Because he can identify these issues. He did a talk on "Database Systems: Road Kill on the Information Superhighway?[2]". Perfect for telling us how we missed the boat in terms of the web. The web is a big database, but that's not how it's viewed.

That's how we view it, but that's not how the world views it.

That's right. We are road kill on this.

Okay, reminds me of the fact that you're in the ACM History Committee.

I'm no longer; I went off about a year ago.

Okay you were on the ACM History Committee. It's now history that you were on the ACM History Committee and computer science is all about, in my mind, inventing the future. So what role does history play in a community that is all about creating a new version of history, so to speak?

I have a couple of different responses. One is that, as I said, databases is fifty years old. That means that the people that started it are getting very old. So we're about to lose a lot of our history. The history committee has commissioned a lot of interviews with these early pioneers. For instance, Charlie Bachman was interviewed by SIGMOD[3]. SIGMOD paid for that interview by a professional historian. Everyone should read his interview on the ACM Digital Library. It talks about IDS (his original system), which was an amazing system that has a lot of similarities with the most recent systems. It was a main memory database system that used virtual memory, for instance. Pretty amazing. We wouldn't have that without a history committee in ACM. We need to grab our history before it goes away. We're always looking into the next five years. So we're going to be losing this very vibrant history that we have. Now we're not like physics and mathematics which has hundreds or thousands of years, but that's good, we can actually capture that. And we can put it in the digital form, using the technology we've invented.

I think we need to re-print Charlie Bachman's interview in the SIGMOD Record.

Well, it's 162 pages long…

162 pages?!

Yes, it was a two-day interview. I have to tell a quick story. I got a call a couple years ago from Charlie Bachman saying "Why don't you come over to my house? I just finished the interview". So I said, "Well, where do you live?" It turns out he lived a mile and a

[2] Keynote on VLDB 1995

[3] Charles W. Bachman interview: September 25-26, 2004; Tucson, Arizona. In: Proceedings of the ACM Oral History Interviews, 2006. DOI: 10.1145/1141880.1141882.

half from my house in Tucson, at that time. So I went over there and Tom Haigh was there. He had just finished the second day of the interviews. Charlie walked up and said, "Would you like some nuts?", and held out a little tray. I realized that that was the Turing Award Bowl...

Whoa!

So I got to get a nut from the Turing Award Bowl from Charlie Bachman. But you mentioned Jim Gray. Jim Gray is no longer with us. We didn't ever interview him. So that's a loss. And Ted Codd is no longer with us. So what you were doing through these interviews, with the other people you were interviewing, is going to be very valuable 20 or 30 years in the future, as well as now.

I decided to go back to the first ones and see what people were predicting and how much of it has come true, so we can compare our historical predictions against reality…

I bet you we are pretty bad at that.

I don't know. I don't know yet, I'll check it out.

So speaking of changing overtime, how has ACM's publications changed over time?

Oh excellent question. If you go back, say, 14 years… at this conference in 1998[4], I think that was in Philadelphia[5]. So if you went to that conference, you got a bound printed version of the proceedings. You took that and the other ones you went to at that time, and you put them on your shelf. I'm sure you had a shelf full of proceedings because that was how you got papers. If you didn't have it in your office you had to go down to the library to get it -- fourteen years ago. So SIGMOD, our community, decided to scan all of those proceedings and put them in digital PDFs. Not only that, they talked to all the other database societies and helped pay for, but also encouraged them to scan theirs. Five years later, SIGMOD gave to all of its members the ACM SIGMOD Anthology[6], two DVDs with 150,000 pages of database papers. Then SIGMOD went to ACM and said, "Other SIGs, you should do the same thing". So the SIGs paid for digitizing the entire past history of all of their conferences and journals. That formed the ACM digital library. Then IEEE (we had already done Data Engineering because SIGMOD worked with them) decided to digitize all of their past. So because of SIGMOD and the SIGMOD dues, which helped pay for this, all of the computer science papers are now digitized.

Well you would like that if for no other reason than its history, but if you look at the accesses to the library, which I've never done, how much do people look at that older stuff?

I'm not sure. I think that they would learn a lot by going back further than the last few years in their areas. They need to do directive searches. You don't just pick up a paper and read it. For the very specific things they are working on, it would be useful. For instance, I was reading Charlie Bachman's interview and I sent a note to my PhD student saying he's doing something that you are doing, you should reference it in your dissertation to give some historical context. The next step though from just digitizing is search and of course Google gives us full search through Google Scholar, as do others, like Microsoft. So we can use this and we can get access so much easier than when you and I were doing it fourteen years ago when we had to go to the library. If the library didn't have it we'd have to ask them to buy it, which would take weeks. Now it's available in seconds. So there is really no excuse for not having an accurate and complete provenance on the ideas that you're working on.

That brings me to a related question. So scientists complain that in Computer Science we don't cite things correctly. So for example, I've heard that every paper that we write about databases should be citing the original paper by Codd. We just don't do that. So should we? Or are they just talking from their own perspective?

I don't know how useful it is for everyone to cite Codd. I think that it is very important to put each person's work in context, and that's bigger than the last two years. But I think a more insightful citation, where you say "these are the important precursor ideas" and these are the best places where each of those ideas is described, I think would be the most efficient. So I guess I disagree with the scientists.

It also reminds me of the idea that there's nothing new under the sun. In fact, when Codd proposed relational databases it was actually the third time they've been proposed. The first I think being von Neumann in 1945 or something. But maybe our ability to ignore history and the fact that it failed all these previous times

[4] Recall that this interview was recorded in 2012.

[5] SIGMOD was held in Philadelphia one year later, in 1999.

[6] For more information of the SIGMOD Anthology, please go to http://www.sigmod.org/publications/anthology/

enables us to keep trying the same things and then finally the third time it would work…

I'm not sure they're the same things. They're probably the core of the idea, the kernel of the idea, but it's how you place it in the current context which determines whether or not it is going to be accepted. So, I had a colleague who said "I never read any of the research because I don't want to be biased. I want to do my own thing". I thought, how inefficient can that be? Because you're re-inventing the wheel that other people have already invented. So I think one can go too far and just spending all of one's time reading and one can go too far and not read anything. It requires some real skill to figure out where that middle ground is.

So you had 6 undergrads involved in your research projects last year under funding from NSF's Research Experiences for Undergraduates program. Why bother to write those grant proposals?

So number one, they're really easy to get. They fund almost all of them because they don't get very many. So if you want a few extra undergraduates, go for it. I think that six was too many for me. I think three would have been just about right. So there are a few tricks. It's important to use your graduate students to help direct them, but I just love working with undergraduates.

Why?

Because they're just starting out research, they don't know what research is. So it's this big, scary, wondrous world. They're not jaded at all, like some graduate students are and a lot of professors are. To them, it's totally new. They really are the future. Also I have to explain things very simply to them. It's hard to explain things simply, but when I understand it, I can, better. So that forces me to do that. Also, I'm a professor and this is my profession. Teaching is a part of it and so I want to find the brightest students and spend time with them.

Maybe I should add that my skeptical question is a little misleading, because we have huge numbers of REU7 students at my own group. Maybe it was a little misleading the way I asked that. I don't want people to get the wrong impression. I think we had five last year.

And they're wonderful to work with, aren't they?

We found it benefits both them and us. In fact, in my research center in Singapore, we've had forty-four interns so far and we've just finished our third year.

Wow, undergraduates?

Some are grads, but let's see…90% are undergrads.

That's incredible, that's wonderful.

It works really well for us. Okay, but this is about you, not about me. So let me ask my next question, which is that, I'm told that at work your door always open. How do you maintain your focus if your doors are always open?

I don't know if I do. I find it very hard to context switch. So I've been working at strategies for doing that. But I find that at the end of the day, the time I spend with my students is the most fun… much more fun than sitting by myself writing a paper, but I think it can get me very scattered and that's another trade-off that I'm still working on.

> ***I know how to make that rabbit run faster, but I don't know how to study that rabbit.***

But it sounds like that since you have the open door policy, in some sense you're benefiting more from the interruptions than…

Actually my door is closed, but I have a policy that you can knock anytime. So if they knock and I'm in a meeting, I'd say, "can I talk to you in a little while?" I find that if the door is actually just open, people walking by is very distracting. I work very hard to manage my physical space for more effectiveness.

Do you have any words of advice for fledging or mid-career database researchers or practitioners?

I have no words of advice for practitioners because I am not one. I have great respect for them; they have their own set of challenges. For mid-career, one word of advice would be to figure out what you are best at (this is related to branding). And to really think about that and to do things that utilize that. Don Knuth once said that he picks problems for which he is the best person in the world to solve those problems, given his background. I think that's a great approach. So you have to really think deeply about what special abilities

[7] Research Experiences for Undergraduate (REU) is an NSF program

you bring, and I think that will also increase the passion, which is really important.

Good advice. Among all your past research, do you have a favorite piece of work?

I do! It's the last book I wrote, on temporal databases. It really encapsulated the coordination framework that I've been developing over the last 20 years. It's three sets of three, and I like that symmetry. So it's different kinds of "time". So there are periods, instances, and intervals. There are three kinds of time in databases: there is valid time, transaction time and bi-temporal. And there are three kinds of queries: current, sequenced, and non-sequenced. All of that kind of came together in the book. It was really satisfying to see those ideas coming out in the new standard, which came out on October 2011.

If you magically had extra time to do an additional thing at work that you're not doing now, what would it be?

So I don't like that question. Let me tell you why specifically I don't like that question. Because it sounds like "what should I have done". You didn't say "should", but my philosophy is that (being a temporal database guy) the only thing that exists is the present.

No, but this is about your future!

That's right. So that's how I like to think of it. What would I do now that I did not do before?

That's right, if you had extra time.

Well, it's not if I have extra time because I'm not going to have extra time. It's what would I emphasize now versus not emphasize in something else. I really want to push ergalics. That's what I'm really focusing on. I'm not doing other things so I can do that.

Okay, if you can change one thing about yourself as a computer science researcher, what would it be?

For the future, what would I do differently as a computer science researcher? I need to learn a lot more about statistics and about the philosophy of science, because I'm not trained in that. I have an undergraduate degree in physics, but after that I was trained to be a computer scientist. So I know how to make that rabbit run faster, but I don't know how to study that rabbit. So that's what I've been focusing on. It's been really fun.

Great. Thanks so much for talking to me today, Rick.

Thank you.

Report on the Second International Workshop on Exploratory Search in Databases and the Web (ExploreDB 2015)

Georgia Koutrika
HP Labs, Palo Alto
koutrika@hp.com

Laks V.S. Lakshmanan
Department of Computer Science, University of British Columbia
laks@cs.ubc.ca

Mirek Riedewald
College of Computer and Information Science, Northeastern University, Boston
mirek@ccs.neu.edu

Mohamed A. Sharaf
School of ITEE, University of Queensland, Australia
m.sharaf@uq.edu.au

Kostas Stefanidis
ICS-FORTH, Heraklion
kstef@ics.forth.gr

1. INTRODUCTION

To make Big Data that is growing in both size and diversity widely accessible, data management and analysis systems have to provide appropriate *exploration services*. An analysis might include structured (relations, tables), semi-structured (XML), and "unstructured" (text) data, linked together through relationships encoded as a graph. Some of the data can be precise, others might be probabilistic [15], e.g., due to measurement error or because it was generated by a statistical model. At the same time, the community of potential users is becoming more diverse as well, ranging from database experts and domain scientists to citizen scientists. These users need system services that help them understand the data and enable them to find relevant information, even if they do not completely comprehend the content and relationships in a complex data collection. This broad goal can be addressed in a variety of ways.

Research in the database community has long been exploring how to simplify the process of composing non-trivial queries, starting with query-by-example [17] in the 1970s. Today many structured data collections can be accessed through Web form interfaces and even keyword search [2, 7], where joins are inferred automatically. Query steering [3, 6, 8] extends the idea of example-based query composition by asking the user to label potential result tuples as (ir-)relevant, a topic covered by one of the keynotes. Then query conditions are automatically derived from the labeled examples. Example-based query composition and modification can be further extended by adding more sophisticated search capabilities that automatically include connected entities and information sources.

Exploration also plays a crucial role when dealing with queries that return too many result tuples, or where expected results are missing—the main topic of the second keynote. For example, why-not [4] and how-to [10] queries are reverse data management approaches that explain or automatically modify a given query if it does not produce the desired outputs. Instead of having the user debug and rewrite a query in a tedious trial-and-error process, the system automatically modifies the query based on examples of missing (or undesirable) query result tuples [16]. Query relaxation techniques have a similar goal for over-constrained queries [9, 12]. An alternative to query relaxation based on examples of missing results is to offer query languages that support imprecise conditions. One option are similarity predicates [11, 13], e.g., searching for cars "like" a given model with a price "near" some value. Another is to allow probabilistic conditions [14], e.g., to express that the user is 80% sure that the entity she is looking for had property X.

For a query returning too many results, *ranking* helps the user explore the most important ones [1, 5]. Its success hinges on the selection or design of an appropriate ranking function. In general, it should capture some natural notion of result relevance, measured based on concepts such as novelty, diversity, and surprise. Ranking functions can be personalized based on historic queries or by requesting user input revealing her preferences. Typically

personalization should be achieved with minimal effort required from the user, as discussed below.

In summary, the field of data exploration is diverse in terms of research directions and potential user base. Hence the ExploreDB workshop intends to bring together researchers and practitioners from different fields, ranging from data management and information retrieval to data visualization and human computer interaction. Its goal is to study the emerging needs and objectives for data exploration, as well as the challenges and problems that need to be tackled, and to nourish interdisciplinary synergies. We summarize the outcomes of the second workshop instance held in conjunction with ACM SIGMOD 2015 in Melbourne, Australia.[1]

2. WORKSHOP OUTLINE

The workshop program consisted of two keynote talks and six peer-reviewed research papers.

2.1 Invited Talks

The first keynote talk titled *"Explore-By-Example: A New Database Service for Interactive Data Exploration"* was given by Prof. Yanlei Diao from the University of Massachusetts at Amherst. Prof. Diao pointed out that while computing power, memory size, and the ability to collect data are growing exponentially, human ability to *understand* data remains practically flat. This "big data, same humans" problem motivates the need for new database services that support automated data exploration. To work effectively with a traditional database management system (DBMS), the user needs to understand the database content well, including structure and meaning of relations, and be able to formally express the exact query to obtain the desired result. For applications and users where this does not apply, a new DBMS service for *interactive data exploration* should have the following features: First, users make sense of the data space via *navigation*, automated by the DBMS. Second, the DBMS interprets user interactions and *learns user interests*, so that it can retrieve all relevant results. Third, both online learning and query processing have *interactive performance*.

Explore-by-example supports this functionality by presenting example tuples to the user in order to obtain feedback about their relevance. Classification models trained based on this feedback drive the process of selecting new samples for additional feedback, as well as the generation of the final SQL query that retrieves a result that includes the relevant samples, but not the irrelevant ones. This approach dramatically changes interaction with the DBMS. The traditional query-cycle consists of query formulation and processing, followed by result review that informs query modification. It is somewhat ad-hoc as the "correct" query predicates are unknown initially, labor-intensive as the user has to review possibly large query results, and resource-intensive as the DBMS executes sequences of queries on big data. With explore-by-example, the traditional query-cycle is replaced with a new cycle that starts with labeling of samples as (ir-)relevant, followed by training of a classification model that informs the choice of another set of samples.

Key research challenges revolve around capturing user interest with high accuracy, minimizing user effort for labeling samples, and keeping user wait time acceptable. A decision-tree based algorithm for identifying hyper-rectangular relevant areas in multi-dimensional space performed well in experiments, requiring a few hundred samples to home in on the target regions. User wait time ranged from 1 to 6 seconds, which Prof. Diao considers acceptable. Interestingly, larger database size did not result in larger required sample size, indicating that the approach scales well to big data. A preliminary user study involving seven CS majors familiar with SQL indicated significant reduction in user effort and exploration time.

While successful for linear predicates (i.e., hyper-rectangular regions), dealing with more general predicates significantly increases complexity. Prof. Diao discussed remaining research challenges related to convergence with a minimum number of labeled samples, DBMS optimizations to minimize user wait time, automatic learning of user profiles, more general queries including join and aggregation, and visualization.

In the second keynote, titled *"Principled Optimization Frameworks for Query Reformulation of Database Queries"*, Prof. Gautam Das from the University of Texas at Arlington focused on solutions for the many-answers and the empty-answers problems. He proposed to address both problems through *ranked retrieval*. In particular, when a query is too selective (empty-answer problem), the user can be steered to "partially matching" tuples. And when a query is not selective enough (many-answers problem), she might be steered to the "top-ranked" tuples. In both scenarios, an appropriate

[1]For a summary of the first instance of ExploreDB, please refer to "Georgia Koutrika, Laks V. S. Lakshmanan, Mirek Riedewald, Kostas Stefanidis: Report on the First International Workshop on Exploratory Search in Databases and the Web (ExploreDB 2014). SIGMOD Record 43(2): 49-52 (2014)."

ranking approach is needed.

For the many-answers problem, Prof. Das discussed dynamic faceted search, which suggests additional constraints by presenting values for attributes from the database schema. By picking a value, the user refines the query. Suggestions are ranked based on the objective of minimizing user effort, which is measured in terms of the number of additional query conditions considered by the user before reaching the entity of interest. This ranking problem can be solved by finding an appropriate fully-grown decision tree with minimum expected height.

Similarly, query relaxation suggestions for the empty-answer problem can be ranked based on the objective of minimizing user effort. Intuitively, the system should suggest relaxations that are likely to be accepted by the user and that will steer her toward minimum effort. Prof. Das presented a probabilistic framework for achieving this goal, which relies on estimates for the probability that the user believes a tuple exists in the database and for the likelihood that the user will prefer a tuple in the answer of a relaxed version of the query. An optimal precise and a faster approximate algorithm find the top-ranked relaxations.

2.2 Paper Presentations

The six talks of the technical program covered a variety of issues related to exploratory data analysis, ranging from personalization for query result presentation to complex event processing.

In *"Data Like This: Ranked Search of Genomic Data"*, V.M. Megler, David Maier, Daniel Bottomly, Libbey White, Shannon McWeeney and Beth Wilmot presented their vision "to make searching for data as easy for scientists as searching the Internet." To this end, they proposed ideas for adapting ranked search to big genome data, which contains position-indexed annotations that are a mix of numeric, ordinal, and binary data types. A major challenge is to find and compare different regions based on their similarity of annotations. Indexing and summarization techniques were proposed to achieve acceptable interactive performance.

Query personalization through preferences was explored in *"Unifying Qualitative and Quantitative Database Preferences to Enhance Query Personalization"* by Roxana Gheorghiu, Alexandros Labrinidis and Panos Chrysanthis. A graph-based framework enables the user to specify both qualitative (i.e., which tuple is preferred over the other in a given pair) and quantitative (i.e., a numerical score for each tuple) preferences. These preferences *together* are leveraged for ranking of database tuples, based on a newly introduced notion of preference "intensity."

Xiaoyu Ge, Panos Chrysanthis and Alexandros Labrinidis (*"Preferential Diversity"*) explored how to achieve personalization through preferences on result *diversity*. Since diversity's goal of reducing redundancy can be in conflict with ranking based on relevance, the proposed approach lets the user control the tradeoff between the two. An iterative algorithm then efficiently processes the data, repeatedly selecting the most relevant records and eliminating others similar to them.

Diversity was also the focus in *"Diversifying with Few Regrets, But too Few to Mention"* by Zaeem Hussain, Hina Khan and Mohamed Sharaf. To balance the tradeoff between maximizing diversity and minimizing regret, which measures loss in utility, a hybrid objective function is proposed. The approach distinguishes between *preference* dimensions, for which regret is minimized, and *neutral* dimensions, for which diversity is maximized. The hybrid objective function is a linear weighted combination of the diversity and regret objectives. A greedy heuristic and an algorithm based on local search find solutions efficiently.

Chen Zhang, Rui Meng, Lei Chen and Feida Zhu (*"CrowdLink: An Error-Tolerant Model for Linking Complex Records"*) proposed a new probabilistic model to better leverage crowdsourcing for record linkage, i.e., the task of finding records that refer to the same entity across different data sources. Questions are selected with the goal of minimizing monetary cost. The algorithm is designed for robustness to errors in the workers' answers.

Tatsuki Matsuda, Yuki Uchida and Satoru Fujita (*"Method of Complex Event Processing over XML Streams"*) argued that complex event processing (CEP) can play a major role in exploratory analysis. As events are detected, they can interrupt an exploration process and affect its direction in realtime. To support a wide variety of applications, they focus on streams of XML data. High performance is achieved by optimizing visibly pushdown automata (VPA) used to execute queries.

3. WORKSHOP CONCLUSIONS

Several themes emerged in the discussions.

- Dealing with large query results is a promising direction for exploratory search. Many meaningful and natural ranking approaches have been proposed, but they are often in conflict with each other. For example, many highly

relevant results might be very similar to each other, resulting in low diversity if they all are presented at the top. More research is needed to be able to combine these ideas into frameworks where the user can customize the ranking function based on desired properties.

- Personalization plays a crucial role for exploration of Big Data. Research challenges revolve around the central issue of user effort, in particular how to learn a personalized ranking function with minimal user input or easy-to-obtain input. For example, it might be easy to label individual records as (ir-)relevant, but it would be practically impossible to expect a user to specify an explicit ranking function.

- The curse of dimensionality is further underscored in Big Data exploration. In particular, guiding users through an uncharted high-dimensionality data space increases the complexity of the data exploration process and challenges its effectiveness. The impact of dimensionality is equally emphasized when ranking a query result, or refining and steering imprecise queries. Hence, it is essential to integrate emerging data exploration techniques with effective methods for handling high-dimensional data.

- System performance, in particular response time experienced by the user, remains a major challenge for exploratory search. Traditional database approaches for indexing, materialization, and data reduction need to be extended and customized for exploratory search on Big Data.

This second instance of ExploreDB made clear that a lot of research work still needs to be done in the general area of exploration for Big Data. Given the growing interest in industry and academia, we are looking forward to the next instance of this workshop.

4. REFERENCES

[1] S. Agrawal and S. Chaudhuri. Automated ranking of database query results. In *Proc. CIDR*, pages 888–899, 2003.

[2] S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: A system for keyword-based search over relational databases. In *Proc. ICDE*, pages 5–16, 2002.

[3] U. Cetintemel, M. Cherniack, J. DeBrabant, Y. Diao, K. Dimitriadou, A. Kalinin, O. Papaemmanouil, and S. B. Zdonik. Query steering for interactive data exploration. In *Proc. CIDR*, 2013.

[4] A. Chapman and H. V. Jagadish. Why not? In *Proc. ACM SIGMOD*, pages 523–534, 2009.

[5] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum. Probabilistic information retrieval approach for ranking of database query results. *ACM Transactions on Database Systems (TODS*, 31(3):1134–1168, 2006.

[6] K. Dimitriadou, O. Papaemmanouil, and Y. Diao. Explore-by-example: An automatic query steering framework for interactive data exploration. In *Proc. ACM SIGMOD*, pages 517–528, 2014.

[7] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In *Proc. VLDB*, pages 670–681, 2002.

[8] M. S. Islam, C. Liu, and R. Zhou. A framework for query refinement with user feedback. *J. Syst. Softw.*, 86(6):1580–1595, 2013.

[9] U. Junker. QUICKXPLAIN: Preferred explanations and relaxations for over-constrained problems. In *Proc. AAAI*, pages 167–172, 2004.

[10] A. Meliou, W. Gatterbauer, and D. Suciu. Reverse data management. *PVLDB*, 4(12):1490–1493, 2011.

[11] A. Motro. VAGUE: A user interface to relational databases that permits vague queries. *ACM Trans. Inf. Syst.*, 6(3):187–214, 1988.

[12] D. Mottin, A. Marascu, S. B. Roy, G. Das, T. Palpanas, and Y. Velegrakis. A probabilistic optimization framework for the empty-answer problem. *Proc. VLDB Endow.*, 6(14):1762–1773, Sept. 2013.

[13] U. Nambiar and S. Kambhampati. Answering imprecise queries over autonomous web databases. In *Proc. ICDE*, pages 45–54, 2006.

[14] B. Qarabaqi and M. Riedewald. User-driven refinement of imprecise queries. In *Proc. ICDE*, pages 916–927, 2014.

[15] D. Suciu, D. Olteanu, C. Re, and C. Koch. *Probabilistic Databases*. Morgan & Claypool, 2011.

[16] Q. T. Tran and C.-Y. Chan. How to ConQueR why-not questions. In *Proc. ACM SIGMOD*, pages 15–26, 2010.

[17] M. M. Zloof. Query-by-example: The invocation and definition of tables and forms. In *Proc. VLDB*, pages 1–24, 1975.

Dissertation Research Problems in Data Management and Related Areas

Gerard de Melo
IIIS
Tsinghua University
Beijing, China
gdm@demelo.org

Mouna Kacimi
Faculty of Computer Science
Free University of Bozen-Bolzano
Bozen-Bolzano, Italy
Mouna.Kacimi@unibz.it

Aparna S. Varde
Department of Computer Science
Montclair State University
Montclair, NJ, USA
vardea@montclair.edu

ABSTRACT

Databases and related fields such as Information Retrieval, Data Mining and Knowledge Management offer many topics of interest for dissertation research. Specific areas include, for instance, big data, social networks, Web question answering and interactive knowledge discovery. In this article, we provide a summary and critique of research problems presented in these and related areas at a workshop on dissertation proposals and early doctoral research.

Keywords

Wikipedia, IR Evaluation, Exploratory Search, Query Processing, Recommender Systems, Rule Mining

1. INTRODUCTION

New research trends are often best observed in the research topics of doctoral candidates, who benefit from the experience of their advisors but add a fresh perspective. Doctoral consortia, or PhD workshops, have emerged as useful forums for dissemination of student research at an early stage in the course of a PhD. They serve the purpose of enabling young scholars to solicit feedback from world-renowned experts and to publish dissertation proposals and initial research results, which can be indicative of emerging challenges and directions in the community. The PhD Workshop in Information and Knowledge Management (PIKM) has been co-hosted with the ACM Conference on Information and Knowledge Management (CIKM) ever since 2007. PIKM 2014, the 7th Edition, was collocated with CIKM 2014 in Shanghai, China [1]. This article presents the outcomes of this workshop.

The PIKM 2014 workshop had a regular paper track and a short paper track, both with oral and poster presentations. This was in order to increase interaction between the presenters and the audience. A notable highlight of PIKM 2014 was a special track with invited talks and papers by more experienced researchers in addition to a keynote speaker, providing additional guidance and advice to early PhD students.

The keynote speaker was Iadh Ounis, faculty member at the University of Glasgow, UK, who spoke about creating and refining PhD Thesis Statements. Among other points, he forcefully argued that a thesis is meant to spark debate and should thus include statements that could potentially raise further questions. He emphasized that instead of including statements of fact, a thesis should include statements that arouse curiosity, thereby propelling readers to study the dissertation in detail and also inspiring future research in interesting sub-problems emerging from the dissertation. This talk was found extremely useful to PhD students who received practical advice for writing and polishing their dissertation.

The best paper award went to Arunav Mishra from Max Planck Institute for Informatics, Germany for his work on "Linking Today's Wikipedia and News from the Past". This is summarized in Section 3 and more details can be found in the PIKM proceedings [1]. In recent years, PIKM has also been announcing an award for the best reviewer to recognize outstanding contributions by a PC member. The best reviewer for PIKM 2014 was Fabian Suchanek from Télécom ParisTech, France. The program committee team consisted of 23 reviewers from across the globe, spanning 16 countries and 6 continents, with a healthy mix of academia and industry.

Considering these highlights of PIKM 2014, we now present a summary and critique of the research contributions in the forthcoming sections. Section 2 covers invited papers, the topics being social network recommendation methods, interactive mining for local and global association rules and knowledge base rule mining respectively. The slides for these invited talks are available online[1]. Section 3 focuses on regular papers in the areas of Wikipedia and news, evaluation methods, search with modeling and efficient query processing. Section 4 deals with short papers, the two

[1] http://iiis.tsinghua.edu.cn/~weblt/pikm2014/

themes being question answering and outlier detection. Further details on all of this research are available in the PIKM proceedings [1]. Finally, Section 5 describes conclusions and ongoing work.

2. RECOMMENDERS & RULE MINING

2.1 Recommendations in Social Networks

Richi Nayak's invited talk focused on the highly topical issue of recommendation in online dating portals [2]. Conventional recommendation engines work in one direction, recommending objects to users based on their interests. In social networks, however, the interest needs to be mutual, so a form of two-way recommendation is needed. This is specifically challenging in online dating platforms, where some users may enter very specific requirements, perhaps even an ideal "Prince Charming" that no real person in the database can live up to, while others just provide broad categories like "blonde hair" or a popular kind of music taste, which could match many thousands of candidate profiles.

Nayak, a faculty member at Queensland University of Technology, Australia, addresses this issue by selecting different recommendation strategies based on how people are using the platform, distinguishing highly active users from infrequent posters, for instance [2]. These strategies can account for patterns observed in user profile information as well as in user activity logs. As a preprocessing step, co-clustering is used to improve the scalability of the recommendation engine.

2.2 Interactive Mining

As an ABD candidate looking forward to his PhD, Abhishek Mukherji, from Samsung Research, USA (in joint work with Elke A. Rundensteiner and Matthew O. Ward from WPI, USA), discussed results on interfaces that enable association rule mining to be conducted in an interactive manner [3]. Association rules capture salient correlations between items in a data source, e.g. "people who buy dips (tend to) also buy chips". Rule mining has a long history and analysts frequently study such rules in order to improve their business.

In practice, however, this can be very tedious without the right tools, often due to dependencies between rules (e.g. one being a special case of another) resulting in countless near-duplicates and due to different levels of confidence and statistical support. Mukherji et al. proposed new techniques and user interfaces that make this process much easier for the analyst. So-called local patterns, which apply only to specific subsets of the data, are a particular focus in his work [3]. For instance, the analyst might be interested in salary trends that only appear in a particular geographic region and demographic. Efficient algorithms are necessary in order to be able to compute relevant rules in a short amount of time and facilitate interactive exploration without long waiting times. In his recent work at Samsung Research, Mukherji is applying similar techniques to mine interesting patterns of mobile device usage.

2.3 Rule Mining in Knowledge Bases

Luis Galárraga is a doctoral student at Télécom ParisTech, France, and has already published several top papers, including the Best Student Paper at WWW 2013 [4]. His research considers rule mining on collections of knowledge about the world. One might discover, e.g., a rule stating that a person is likely to live in the same city as their spouse. Such a rule can be interesting in itself, or could be used to fill the gaps when information is missing in a database. This is an important task because even the largest available knowledge bases are known to be very incomplete.

Galárraga's research proposes novel techniques to assess the confidence of rules in this setting, overcoming some of the problems of the traditional closed world assumption, according to which any knowledge not in the database is assumed to be false. This assumption cannot hold true in large open-domain knowledge bases. Galárraga thus proposes the alternative Partial Completeness Assumption. Moreover, he presents scalable techniques to find such rules in very large knowledge collections, yielding results on big popular knowledge bases such as YAGO2 and DBpedia in mere minutes. The same method can also be used to connect different knowledge sources, even when these connections are more complex than mere one-to-one alignments.

3. WIKIPEDIA, EVALUATION, SEARCH AND QUERYING

3.1 Wikipedia and News

To increase user satisfaction about the results of Information Retrieval systems, an interesting approach was proposed by A. Mishra. This aimed at combining different information sources to enrich knowledge about events. More specifically, it focused on Wikipedia and news articles, which provide different levels of description about events [1]. While Wikipedia excerpts describe events in an abstract form omitting details, news articles may describe events in an overly detailed form, missing the overall picture. Thus, the goal was to combine these two sources by creating a link from any Wikipedia excerpt to a matching set of news articles and vice versa. The proposed approach modeled the problem as an IR problem. It exploited two text collections, the first one being a collection of news articles and the second one a collection of Wikipedia excerpts. For the first corpus, the query was a Wikipedia excerpt and for the second one the query was a news

article. The authors demonstrated that unrelated Wikipedia excerpts and news articles may use the same vocabulary, and thus a keyword-based retrieval strategy delivered only mediocre results. To overcome this, they developed a new strategy that added timestamps to both Wikipedia excerpts and news articles. These timestamps were used to compute a distribution of time expressions in the top-k documents and then re-rank the entire result list by boosting those that have similar time expressions. Future work on text mining and entity resolution was considered by the authors to improve the quality of the results.

3.2 Robust and Reusable Evaluation

The importance of understanding a user's information need to improve the quality of exploratory search was emphasized in the paper by K. Athukorala [1]. The main challenge of this research was that user knowledge and needs changed as the search progressed, which required adequate prediction of relevant results to evolving user intents. The authors approached this problem by making an exploratory study of the behavior of academics in searching information. This application captured the essence of exploratory search, since scientific searches often dealt with the discovery of unfamiliar topics. The authors developed a formal model to represent the state of exploration using observable aspects of user behavior, including viewed search results and clicking actions. This model was then used to predict the relevance of search results to current user interests and knowledge. Further steps were considered to improve the prediction power of such a model by exploiting other implicit interaction data e.g., read-time, click-time, scroll length, and gaze distribution over results.

3.3 Exploratory Search through Modeling

The author K. Hui focused on the evaluation of Information Retrieval systems [1]. Currently, the evaluation of such systems is performed through manual assessment, where the result documents are labeled to indicate their degree of relevance for the query. These labels, however, are associated to the entire document and do not correspond to its content. Consequently, manual assessment can hardly be extended to unlabeled parts of the document collection. Moreover, it is very expensive and cannot be applied to large scale datasets. To address this problem, the author presented a new evaluation strategy for diversity and novelty of search results. The proposed approach connected the evaluation results to the content of documents. It generated, for each sub-topic, a ground truth language modeled from a set of sufficient labeled documents. Thus, evaluation results could be re-used to assess future information retrieval systems even when human labeling was not possible.

3.4 Efficient Query Processing

Uysal et al. addressed the problem of efficient query processing in Information Retrieval systems that performed similarity search of multimedia content [1]. For that purpose, the authors considered a distance measure known as the Earth Mover's Distance (EMD). This distance measure assesses image dissimilarity in terms of the minimum amount of work needed to transform one feature representation into another one. The main advantage of this distance measure was its strong expressiveness of perceptual similarity and its applicability to both feature histograms and signatures. A major impediment to using this distance measure, however, had been its exponential time complexity with respect to increasing numbers of representatives. The authors focused on how to reduce the complexity of EMD after presenting the main challenges related to efficient query processing on feature signatures. They proposed a new lower bound Independent Minimization for Signatures (IM-Sig) to the EMD on feature signatures. This lower bound was regarded as an efficient filter approximation approach combined with k-nearest neighbor queries. The authors presented extensive experiments showing highly efficient results of the proposed approach.

4. QUESTION ANSWERING AND OUTLIER DETECTION

4.1 Question Expansion in QA Services

This paper was presented by Kyoungman Bae and Youngjoong Ko from Dong-A University, Busan, South Korea. It detailed a question expanding method to classify questions for question-answering (QA) services [1]. Input questions are mostly written with just a small portion of text, and, due to this fact, may not always give sufficient details for good classification. The authors thus proposed to expand the questions as follows. They obtained question-answer pairs pertaining to an input question with a search engine and selected top relevant words for expansion. They then generated pseudo answers adding question-related words using translation probabilities from questions to answers. Their preliminary experiments indicated that QA services provided better answers with this question expansion method.

4.2 Outlier Detection in Subspaces

Researchers Zhana Bao and Wataru Kameyama from Waseda University in Tokyo, Japan presented a novel outlier detection method. The authors explained that current methods find prominent outliers but neglect certain kinds of hidden ones [1]. The authors instead proposed a two-stage inspection model to detect outliers in different subspaces. The first stage measured neighboring density in subspaces to discover low

dimensional outliers. The second stage assessed the degree of deviation of neighbors in joint subspaces. The authors statistically analyzed the results, merging them into a single score for each item, and candidate outliers were output as top-scoring objects. This work was evaluated on both synthetic and real data sets and was proven to be better than existing methods.

5. CONCLUSIONS

We observe a continued trend for young researchers to investigate Data Management issues arising in more specific settings and domains. Examples include news retrieval, user modeling, data mining, knowledge bases, and online dating. This emphasizes the importance of multidisciplinary work spanning Data Management that has extended its horizons to many fields within as well as beyond Computer Science.

Much of the work presented at PIKM presents significant potential for future research as well. For example, social network mining for online dating can be further optimized to include criteria such as minimizing search time or reducing the number of unsuccessful hits. News retrieval can be further enhanced by mining data on current trends to find the hot topics that interest specific user communities and displaying these in search engines. Web personalization can be conducted based on user modeling, thus providing better service to users in various applications such as product marketing.

The PIKM workshop provides an excellent forum for presentation of research ideas in early doctoral work. This has been a highly successful event since 2007. The organizers try to introduce interesting aspects to this workshop year after year. For example, the poster track was introduced in 2008, best reviewer awards have been given in some of the recent PIKMs, and this year we had a track with invited papers that included a mix of recent and experienced researchers to motivate early PhD students in several areas. The presenters of the invited papers were in addition to the keynote speaker. The keynote track has been in PIKM for quite a few years now and we have many prominent speakers give us very exciting and inspiring talks on topics that are useful to PhD students, over and above presenting their own research for further inspiration.

We sincerely hope that PIKM continues to be an important highlight of CIKM every year. This workshop certainly encourages PhD students to present their dissertation proposals and early doctoral research. It serves the dual purpose of publishing their work and getting feedback from a worldwide audience. It also helps meeting fellow students and researchers for collaborative opportunities, job prospects and friendships. Finally, it provides a unique perspective on research topics that are likely to grow in importance in data management and related areas.

6. ACKNOWLEDGMENTS

The authors would like to thank the CIKM 2014 organizers for giving us the opportunity to chair this PhD workshop. We convey sincere thanks to the keynote speaker Iadh Ounis and the presenters of the invited papers, Richi Nayak, Abhishek Mukherji and Luis Galárraga.

We express our gratitude towards all the PC members of the PIKM 2014 workshop for their special efforts in reviewing papers and thereby guiding young researchers based on their expertise.

Gerard de Melo's research is supported by China 973 Program Grants 2011CBA00300, 2011CBA00301, and NSFC Grants 61033001, 61361136003, 61450110088.

Aparna Varde's participation in PIKM was supported by funds from the CSAM (College of Science and Math) Dean's Office at Montclair State University, NJ, USA.

7. REFERENCES

[1] Gerard de Melo, Mouna Kacimi, Aparna S. Varde (Eds.): Proceedings of the 7th Workshop on Ph.D Students, PIKM at CIKM 2014, Shanghai, China, November 3, 2014. ACM, ISBN 978-1-4503-1481.

[2] Lin Chen, Richi Nayak. Leveraging the network information for evaluating answer quality in a collaborative question answering portal. Social Network Analysis and Mining 2(3), pp. 197-215, Springer, 2012.

[3] Abhishek Mukherji, Elke A. Rundensteiner, and Matthew O. Ward. COLARM: Cost-based optimization for localized association rule mining. In Proceedings of EDBT, pages 181–192, 2014.

[4] Luis Galárraga, Christina Teflioudi, Katja Hose, Fabian Suchanek. Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases. In Proceedings of WWW, pages 413-422, 2013.

How Connected are the ACM SIG Communities?

Fabricio Benevenuto
Federal University of Minas Gerais, Brazil
fabricio@dcc.ufmg.br

Alberto H. F. Laender
Federal University of Minas Gerais, Brazil
laender@dcc.ufmg.br

Bruno L. Alves
Federal University of Minas Gerais, Brazil
bruno.leite@dcc.ufmg.br

ABSTRACT

Currently, computer scientists publish more in conferences than journals and several conferences are the main venue in many computer science subareas. There has been considerable debate about the role of conferences for computer science research and one of the main arguments in favor of them is that conferences bring researchers together, allowing them to enhance collaborations and establish research communities in a young and fast-evolving discipline. In this work, we investigate if computer science conferences are really able to create collaborative research communities by analyzing the structure of the communities formed by the flagship conferences of several ACM SIGs. Our findings show that most of these flagship conferences are able to connect their main authors in large and well-structured communities. However, we have noted that in a few ACM SIG flagship conferences authors do not collaborate over the years, creating a structure with several small disconnected components.

1. INTRODUCTION

There is a long debate about the role of conference publications in computer science [3, 13–16]. On one hand, some researchers argue that conferences offer a fast and regular venue for publication of research results at the same time that allow researchers to interact with each other. These interactions would be the key for the development of research communities in a relatively young and fast-evolving discipline. On the other hand, there exists some criticism to the conference system due to the short time given to review the papers, the limited size of the papers, the review overload faced by program committee members, and the limited time for authors to revise their papers after receiving the reviews.

Despite the existing concerns on this controversial issue, conferences are quite important today as computer scientists give a huge value to them [4, 6, 9]. Particularly, the flagship conferences of the ACM Special Interest Groups (SIGs) are often the most prestigious ones, usually being listed among the most important venues of several computer science subareas.

Although the importance of the main ACM SIG conferences to their respective research fields is incontestable, part of the argument in favor of conferences is that they help create and maintain an active research community, by simply offering a place for researchers to meet regularly and promote collaborations. In this work, we aim at investigating two questions related to this context: (1) *How structured are the ACM SIG conference communities?* and (2) *Who are the individuals responsible for connecting each ACM SIG conference community?*

Our effort to answer the first question consists in analyzing the coauthorship graph structure of the communities formed by the flagship conferences of the ACM SIGs. Our findings show that most of the ACM SIG conferences are able to connect their main authors in large and well-structured connected components of a coauthorship network and only very few conferences, such as the ACM Symposium on Applied Computing, flagship conference of SIGAPP, and the ACM Conference on Design of Communications, flagship conference of SIGDOC, do not form the typical structure of a research community, presenting a set of small and disconnected components.

To approach our second question, we present a tool that allows one to visualize research communities formed by authors from specific ACM SIG conferences, making it possible to identify the most prolific authors with a high level of participation in a given community. To do that, we use data from DBLP[1] and Google Scholar[2] to construct scientific communities and identify their leaders. Our visualization tool also allows a plethora of interesting observations about the authors as we shall see later.

[1] http://dblp.uni-trier.de
[2] scholar.google.com

Table 1: DBLP statistics for the flagship conferences of the ACM SIGs

SIG	Acronym	Period	Authors	Publications	Editions	Aut/Edi	Pub/Edi	Aut/Pub
SIGACT	STOC	1969-2012	2159	2685	44	49.07	61.02	0.80
SIGAPP	SAC	1993-2011	9146	4500	19	481.37	236.84	2.03
SIGARCH	ISCA	1976-2011	2461	1352	36	68.36	37.56	1.82
SIGBED	HSCC	1998-2012	846	617	15	56.40	41.13	1.37
SIGCHI	CHI	1994-2012	5095	2819	19	268.16	148.37	1.81
SIGCOMM	SIGCOMM	1988-2011	1593	796	24	66.38	33.17	2.00
SIGCSE	SIGCSE	1986-2012	3923	2801	27	145.30	103.74	1.40
SIGDA	DAC	1964-2011	8876	5693	48	184.92	118.60	1.56
SIGDOC	SIGDOC	1989-2010	1071	810	22	48.68	36.82	1.32
SIGGRAPH	SIGGRAPH	1985-2003	1920	1108	19	101.05	58.32	1.73
SIGIR	SIGIR	1978-2011	3624	2687	34	106.59	79.03	1.35
SIGKDD	KDD	1995-2011	3078	1699	17	181.06	99.94	1.81
SIGMETRICS	SIGMETRICS	1981-2011	2083	1174	31	67.19	37.87	1.77
SIGMICRO	MICRO	1987-2011	1557	855	25	62.28	34.20	1.82
SIGMM	MM	1993-2011	5400	2928	19	284.21	154.11	1.84
SIGMOBILE	MOBICOM	1995-2011	1151	480	17	67.71	28.24	2.40
SIGMOD	SIGMOD	1975-2012	4202	2669	38	110.58	70.24	1.57
SIGOPS	PODC	1982-2011	1685	1403	30	56.17	46.77	1.20
SIGPLAN	POPL	1975-2012	1527	1217	38	40.18	32.03	1.25
SIGSAC	CCS	1996-2011	1354	676	16	84.63	42.25	2.00
SIGSAM	ISSAC	1988-2011	1100	1177	24	45.83	49.04	0.93
SIGSOFT	ICSE	1987-2011	3502	2248	25	140.08	89.92	1.56
SIGUCCS	SIGUCCS	1989-2011	1771	1593	23	77.00	69.26	1.11
SIGWEB	CIKM	1992-2011	4978	2623	20	248.90	131.15	1.90

The rest of this paper is organized as follows. Next section introduces the ACM SIG communities we have considered. Then, we characterize the structure of ACM SIG communities and analyze the role of their leaders. Finally, we conclude by summarizing our results.

2. ACM SIG COMMUNITIES

In order to construct scientific communities from ACM SIG conferences, we have gathered data from DBLP [10, 11], a digital library containing more than 3 million publications from more than 1.5 million authors that provides bibliographic information on major computer science conference proceedings and journals. DBLP offers its entire database in XML format, which facilitates gathering the data and constructing entire scientific communities.

Each publication is accompanied by its title, list of authors, year of publication, and publication venue, i.e., conference or journal. For the purpose of our work, we consider a research network as a coauthorship graph in which nodes represent authors (researchers) and edges link coauthors of papers published in conferences that put together specific research communities [1]. In order to define such communities, we focus on the publications from the flagship conferences of major ACM SIGs. Thus, we define a scientific community by linking researchers that have coauthored a paper in a certain conference, making the ACM SIG flagship conferences to act as communities in which coauthorships are formed.

In total, 24 scientific communities have been constructed. Table 1 lists these communities, including the respective ACM SIG, the conference acronym, the period considered (some conferences had their period reduced to avoid hiatus in the data), the total number of authors, publications and editions as well as ratios extracted from these last three figures. We make this dataset available for the research community. For more details, we refer the reader to our previous efforts that use it [1, 2].

3. STRUCTURE OF THE ACM SIG COMMUNITIES

Ideally, it is expected that over the years conferences are able to bring together researchers with common interests so that they can collaborate to advance a certain field. Thus, it is expected that with a few decades, the coauthorship graph of a certain community contains a largest connected component (LCC) [12] that puts together a large part (i.e., the majority) of its authors. In other words, one could expect a large LCC in a research community in which authors often interact and collaborate, meaning that there exists at least one path among a large fraction of them.

Table 2 shows the percentage of the authors of each community that are part of the largest connected component of its respective coauthorship graph. Clearly, we can note that most of the research communities formed by SIG conferences have a large connected component that is typically larger than half of the network, suggesting that these conferences have successfully put together their researchers in a collaborative network. Figure 1 depicts the networks of the three conferences with the most representative largest connected components, SIGMOD, STOC and CHI, and the three conferences with the least representative ones, SIGUCCS, SAC and SIGDOC. In these networks, connected components are shown with different colours and the LCC is presented

as the most central one. The size of each node represents an estimative of the importance of a researcher to the scientific community, which is discussed in the next section. As we can see, the latter are the only three communities that are formed by a very small largest connected component (i.e., with less than 10% of the researchers in the network) and several other small connected components. Typically, these conferences cover a wide range of topics, making it difficult for their researchers to establish a research community. For example, SAC is an annual conference organized in technical tracks that change at each edition. Although this dynamic format attracts a large number of submissions every year, it does not contribute to the formation of a specific, well-structured research community.

Table 2: Structure of the scientific communities

Conference	Largest Connected Component
SIGMOD	74.75%
STOC	74.34%
CHI	73.33%
MICRO	65.13%
HSCC	62.53%
DAC	62.21%
KDD	61.24%
ISCA	58.72%
SIGCOMM	57.88%
SIGIR	57.86%
SIGCSE	55.31%
ICSE	52.68%
PODC	52.46%
CIKM	51.81%
CCS	51.70%
SIGMETRICS	50.89%
POPL	50.82%
MM	50.06%
SIGGRAPH	46.72%
ISSAC	44.09%
MOBICOM	37.88%
SIGDOC	9.69%
SAC	3.67%
SIGUCCS	3.27%

4. LEADERS AND THEIR ROLES IN RESEARCH COMMUNITIES

We now turn our attention to our second research question related to identifying important members of a research community. Our intention here is not to rank researchers within their communities, but to give a sense about which researchers have being engaged in a certain community for consecutive years and mostly helped connecting its final coauthorship graph. Thus, instead of attempting to quantify centrality measures [5, 7] of authors and node degree in coauthorship graphs, we have defined a metric that aims at quantifying the involvement of a researcher in a scientific community in terms of publications in its flagship conference over the years. Intuitively, this metric should be able to capture (i) the prolificness of a researcher and (ii) the frequency of her involvement with a certain community. Next we discuss how exactly we have defined this metric.

4.1 Quantifying a Researcher's Engagement in a Community

First, in order to capture the prolificness of a researcher, we use the h-index [8], a metric widely adopted for this purpose. This metric consists of an index that attempts to measure both the productivity and the impact of the published work of a researcher. It is based on the set of the researcher's most cited publications and the number of citations that they have received. For example, a researcher r has an h-index h_r if she has at least h publications that have received at least h citations. Thus, for instance, if a researcher has 10 publications with at least 10 citations, her h-index is 10.

Then, as an attempt to capture the importance of a researcher to a specific community in a certain period of time, we multiply her h-index by the number of publications this researcher has in a certain community (conference) during a time window. We name this metric *CoScore*, as it aims to measure the importance of a researcher as a member of the community [1]. More formally, the *CoScore* of a researcher r in a community c during a period of time t, $CoScore_{r,c,t}$, is given by her h-index h_r multiplied by the number of publications r has in c during t ($\#publications_{r,c,t}$), as expressed by the following equation:

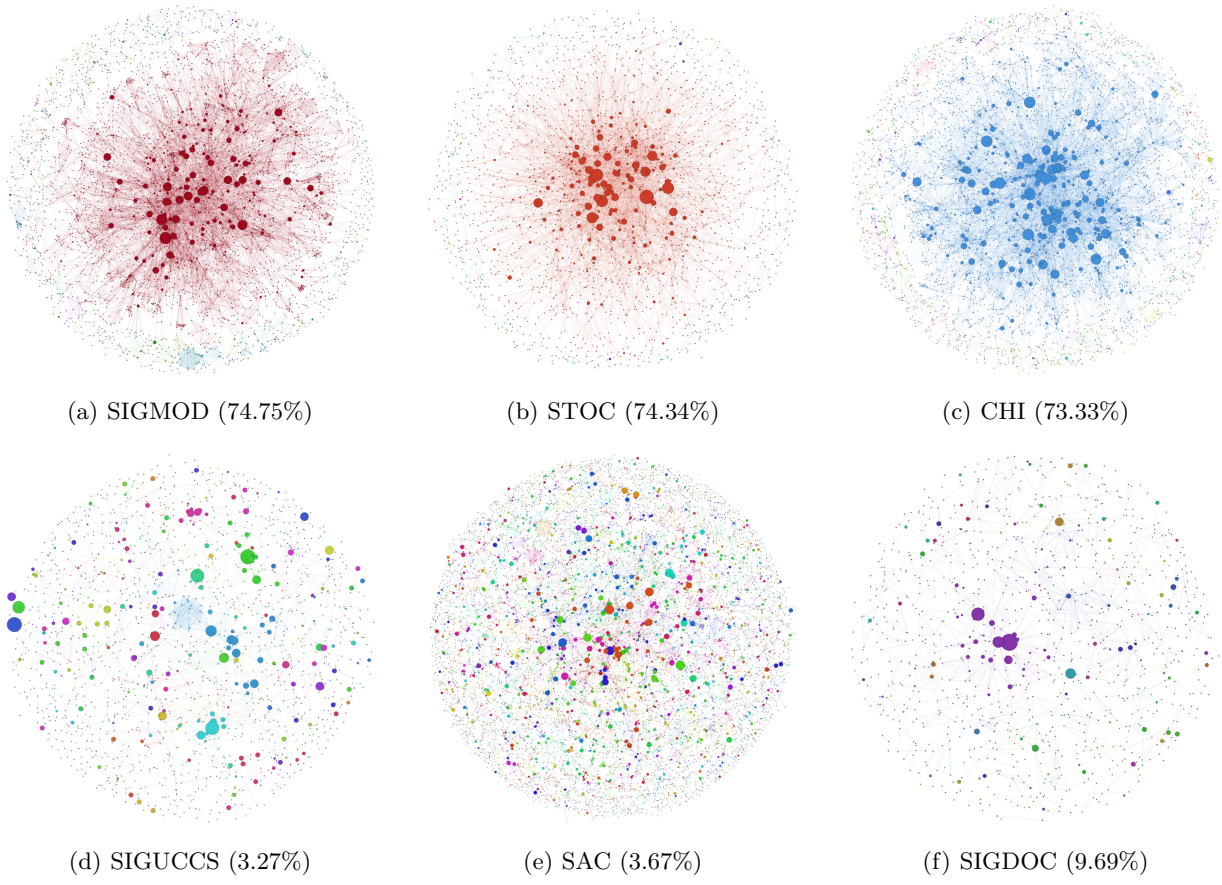
$$CoScore_{r,c,t} = h_r \times \#publications_{r,c,t} \qquad (1)$$

We note that the first part of the above equation captures the importance of a researcher to the scientific community as a whole regardless of any specific research area or period of time, and the second part weights this importance based on the activity of the researcher in a certain community over a period of time. The idea is to compute the amount of time a certain research appeared among the top researchers in terms of this metric over periods of a few consecutive years. For example, if a researcher that today has a high h-index has published four papers at KDD in a period of three years, it means she is engaged with that community at least for that short period of time. If a researcher appears among the top ones within a community for several of these periods, it suggests that she has a life of contributions dedicated to that community. Next, we briefly describe how we have inferred the h-index of the researchers.

4.2 Inferring Researchers' H-index

There are multiple tools that measure the h-index of researchers, out of which Google Scholar Citations[3] is the most prominent one. However, to have a profile in this system, a researcher needs to sign up and explicitly create her research profile. In a preliminary collection of part of the profiles of the DBLP authors, we found that less than 30% of these authors had a profile on

[3]http://scholar.google.com/citations



(a) SIGMOD (74.75%) (b) STOC (74.34%) (c) CHI (73.33%)

(d) SIGUCCS (3.27%) (e) SAC (3.67%) (f) SIGDOC (9.69%)

Figure 1: Scientific communities and the size of their LCC

Google Scholar. Thus, this strategy would reduce our dataset and potentially introduce bias when analyzing the communities.

To divert from this limitation, we used data from the SHINE (Simple HINdex Estimation) project[4] to infer the researchers' h-index. SHINE provides a website that allows users to check the h-index of almost 1800 computer science conferences. The SHINE developers crawled Google Scholar, searching for the title of papers published in these conferences, which allowed them to effectively estimate the h-index of the target conferences based on the citations computed by Google Scholar. Although SHINE only allows one to search for the h-index of conferences, the SHINE developers kindly allowed us to access their dataset to infer the h-index of researchers based on the conferences they crawled.

However, there is a limitation with this strategy. As SHINE does not track all existing computer science conferences, researchers' h-index might be underestimated when computed with this data. To investigate this issue, we compared the h-index of a set of researchers with a profile on Google Scholar with their estimated h-index based on the SHINE data. For this, we randomly selected 10 researchers from each conference in Table 1 and extracted their h-indexes from their Google Scholar profiles. In comparison with the h-index we estimated from SHINE, the Google Scholar values are, on average, 50% higher. Figure 2 shows the scatterplot for the two h-index measures. We can note that although

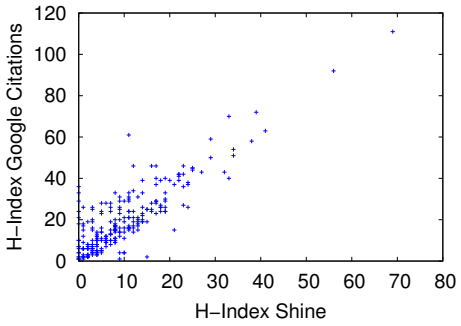


Figure 2: Correlation between the inferred h-index and Google Scholar Citations one

[4]http://shine.icomp.ufam.edu.br/

the SHINE-based h-index is smaller, the two measures are highly correlated. The Pearson's correlation coefficient is 0.85, which indicates that researchers might have proportional h-index estimations in both systems.

4.3 Visualizing Community Members and their Roles within the Communities

In order to make our results public, we have developed an interactive tool[5] that allows one to browse the scientific communities, visualizing their structures and the contribution of each specific researcher to connect their coauthorship graph. Our effort consists in allowing users to search for researchers based on the metric presented in the previous section. The size of each author's node is proportional to the number of times she appears within the top 10% researchers with highest *CoScore* values in a time window of three years. Figure 3 shows, for example, the coauthorship graph of Michael Stonebracker, the winner of the 2014 A.M. Turing Award[6], and his connections within the SIGMOD community. These connections are highlighted when one passes the mouse over the researcher's name. In addition, our tool allows one not only to search for authors but also to visualize statistics about them within the communities.

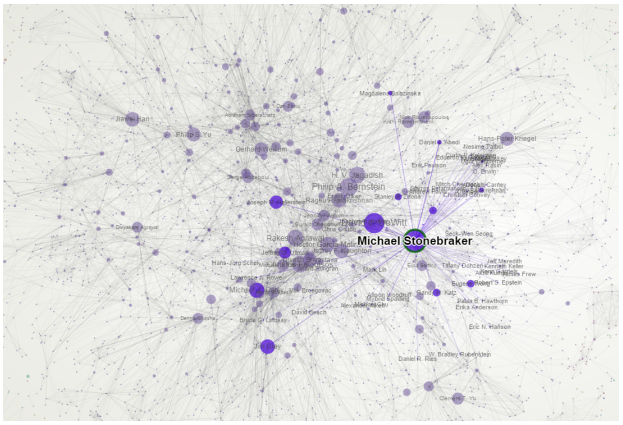


Figure 3: Michael Stonebraker and his connections within the SIGMOD community

To check if our approach really identifies those who are prolific and engaged in a specific community, we notice that several research communities have established different awards to recognize those who were important to a certain field and helped to advance or even build a certain community. Thus, we use some of these awards to corroborate the effectiveness of our metric in establishing the importance of a researcher within a specific community. We have computed a ranking of the researchers that appear most often in the top 10% of the *CoScore* ranking over the years for each community. We have chosen the CHI, ICSE, KDD, POPL, SIGCOMM, SIGGRAPH, SIGIR, and SIGMOD communities to show their top 20 researchers in Tables 3 and 4. As we can see, several well known names appear in these top lists, including past keynote speakers of those conferences and awardees for their life time contributions in the respective community (names in bold). In addition, besides Michael Stonebraker, these top lists include four other winners of the A.M. Turing Award (indicated by asterisks): Amir Pnueli (1996), Jim Gray (1998), Edmund M. Clarke (2007) and Barbara Liskov (2008). Indeed, by analyzing all these awardees from each community, we found that a large fraction of them appeared in the top 10% of the *CoScore* ranking at least once in the conference history. For example, according to the respective ACM SIG websites, these fractions are 75% for KDD[7], 35% for SIGCOMM[8], 60% for SIGIR[9], and 80% for SIGMOD[10]. Except for SIGCOMM, a community with many sponsored conferences that were not considered in our dataset, the other three communities presented very high numbers of awardee members that appear at least once in the top 10% of the *CoScore* ranking over the years. These observations provide evidence that our approach correctly captures the notion we wanted to.

5. CONCLUSIONS

This work analyzes the structure of the communities formed by the flagship conferences of ACM SIGs. Our findings show that most of the ACM SIGs are able to connect their main authors in large and visually well-structured communities. However, we note that a few conferences, such as the ACM Symposium on Applied Computing, flagship conference of SIGAPP, and the ACM Conference on Design of Communications, flagship conference of SIGDOC, do not form a strong research community, presenting a structure with several disconnected components. We have opened our results to the research community as an interactive visualization tool that allows one to browse the scientific communities, visualizing their structures and the contribution of each specific researcher to connect its coauthorship graph.

Acknowledgments

This work was partially funded by InWeb - The Brazilian National Institute of Science and Technology for the Web (grant MCT/CNPq 573871/2008-6), and by the authors' individual grants from CNPq, CAPES e FAPEMIG.

[5]Available at `www.acmsig-communities.dcc.ufmg.br`
[6]`http://amturing.acm.org/stonebraker_1172121.pdf`
[7]http://www.sigkdd.org/awards_innovation.php
[8]http://www.sigcomm.org/awards/sigcomm-awards
[9]http://www.sigir.org/awards/awards.html
[10]http://www.sigmod.org/sigmod-awards

Table 3: Researchers that appear most often in the top 10% of the CoScore ranking over the years

CHI	ICSE	KDD	POPL
Scott E. Hudson	**Victor R. Basili**	**Heikki Mannila**	Thomas W. Reps
Hiroshi Ishii	**Barry W. Boehm**	Hans-Peter Kriegel	Martn Abadi
Steve Benford	**Jeff Kramer**	**Jiawei Han**	John C. Mitchell
George G. Robertson	**Mary Shaw**	Martin Ester	Robert Harper
Shumin Zhai	Dewayne E. Perry	**Rakesh Agrawal**	Zohar Manna
Brad A. Myers	Don S. Batory	Bing Liu	Benjamin C. Pierce
Robert E. Kraut	Mary Jean Harrold	Ke Wang	Amir Pnueli*
Elizabeth D. Mynatt	**Lori A. Clarke**	**Padhraic Smyth**	**Barbara Liskov***
Ravin Balakrishnan	Gruia-Catalin Roman	Philip S. Yu	Martin C. Rinard
James A. Landay	Premkumar T. Devanbu	Charu C. Aggarwal	Luca Cardelli
Ken Hinckley	Gail C. Murphy	**Vipin Kumar**	Thomas A. Henzinger
Mary Czerwinski	**Richard N. Taylor**	Wynne Hsu	**Ken Kennedy**
Carl Gutwin	**David Garlan**	Qiang Yang	**Matthias Felleisen**
Gregory D. Abowd	Michael D. Ernst	**Christos Faloutsos**	Edmund M. Clarke*
Michael J. Muller	James D. Herbsleb	William W. Cohen	Mitchell Wand
Susan T. Dumais	Lionel C. Briand	Pedro Domingos	David Walker
Loren G. Terveen	Gregg Rothermel	Eamonn J. Keogh	Simon L. Peyton Jones
Steve Whittaker	Kevin J. Sullivan	Alexander Tuzhilin	Shmuel Sagiv
W. Keith Edwards	**David Notkin**	Mohammed Javeed Zaki	Barbara G. Ryder
John M. Carroll	Douglas C. Schmidt	Mong-Li Lee	Alexander Aiken

Table 4: Researchers that appear most often in the top 10% of the CoScore ranking over the years

SIGCOMM	SIGGRAPH	SIGIR	SIGMOD
Scott Shenker	**Donald P. Greenberg**	**W. Bruce Croft**	**Michael Stonebraker***
George Varghese	**Pat Hanrahan**	Clement T. Yu	**David J. DeWitt**
Donald F. Towsley	Demetri Terzopoulos	**Gerard Salton**	**Philip A. Bernstein**
Ion Stoica	**David Salesin**	Alistair Moffat	H. V. Jagadish
Hui Zhang	**Michael F. Cohen**	**Susan T. Dumais**	Christos Faloutsos
Deborah Estrin	**Richard Szeliski**	James Allan	**Rakesh Agrawal**
Hari Balakrishnan	John F. Hughes	Yiming Yang	**Michael J. Carey**
Robert Morris	N. Magnenat-Thalmann	Edward A. Fox	**H. Garcia-Molina**
Thomas E. Anderson	**Tomoyuki Nishita**	James P. Callan	Jiawei Han
Ramesh Govindan	**Andrew P. Witkin**	Chris Buckley	Raghu Ramakrishnan
Srinivasan Seshan	Norman I. Badler	**C. J. van Rijsbergen**	Jeffrey F. Naughton
David Wetherall	**Peter Schrder**	Justin Zobel	**Jim Gray***
Yin Zhang	Steven Feiner	Ellen M. Voorhees	Hans-Peter Kriegel
Jennifer Rexford	**Hugues Hoppe**	Mark Sanderson	Gerhard Weikum
Jia Wang	**Jessica K. Hodgins**	**Norbert Fuhr**	Philip S. Yu
J. J. Garcia-Luna-Aceves	**Greg Turk**	Nicholas J. Belkin	Divesh Srivastava
Randy H. Katz	**Marc Levoy**	Chengxiang Zhai	Joseph M. Hellerstein
Albert G. Greenberg	**P. Prusinkiewicz**	Charles L. A. Clarke	Krithi Ramamritham
Mark Handley	Eihachiro Nakamae	Alan F. Smeaton	Nick Roussopoulos
Simon S. Lam	Dimitris N. Metaxas	Gordon V. Cormack	**Surajit Chaudhuri**

6. REFERENCES

[1] B. L. Alves, F. Benevenuto, and A. H. F. Laender. The Role of Research Leaders on the Evolution of Scientific Communities. In *Proceedings of the 22nd International Conference on World Wide Web (Companion Volume)*, pages 649–656, 2013.

[2] F. Benevenuto, A. H. F. Laender, and B. L. Alves. The H-index paradox: your coauthors have a higher H-index than you do. *Scientometrics*, 2016 (to appear).

[3] L. Fortnow. Time for Computer Science to Grow Up. *Commun. ACM*, 52(8):33–35, Aug. 2009.

[4] M. Franceschet. The Role of Conference Publications in CS. *Commun. ACM*, 53(12):129–132, Dec. 2010.

[5] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[6] J. Freyne, L. Coyle, B. Smyth, and P. Cunningham. Relative Status of Journal and Conference Publications in Computer Science. *Commun. ACM*, 53(11):124–132, Nov. 2010.

[7] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002.

[8] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46):16569–16572, 2005.

[9] A. H. F. Laender, C. J. P. de Lucena, J. C. Maldonado, E. de Souza e Silva, and N. Ziviani. Assessing the Research and Education Quality of the Top Brazilian Computer Science Graduate Programs. *SIGCSE Bulletin*, 40(2):135–145, 2008.

[10] M. Ley. The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. In *Proceedings of the 9th International Symposium on String Processing and Information Retrieval*, pages 1–10, 2002.

[11] M. Ley. DBLP: Some Lessons Learned. *Proc. of VLDB Endow.*, 2(2):1493–1500, Aug. 2009.

[12] M. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[13] D. A. Patterson. The Health of Research Conferences and the Dearth of Big Idea Papers. *Commun. ACM*, 47(12):23–24, Mar. 2004.

[14] M. Y. Vardi. Conferences vs. Journals in Computing Research. *Commun. ACM*, 52(5):5–5, May 2009.

[15] M. Y. Vardi. Revisiting the Publication Culture in Computing Research. *Commun. ACM*, 53(3):5–5, Mar. 2010.

[16] M. Y. Vardi. Scalable Conferences. *Commun. ACM*, 57(1):5–5, Jan. 2014.