Probabilistic Databases

Synthesis Lectures on Data Management

Editor
M. Tamer Özsu, ***University of Waterloo***

Synthesis Lectures on Data Management is edited by Tamer Özsu of the University of Waterloo. The series will publish 50- to 125 page publications on topics pertaining to data management. The scope will largely follow the purview of premier information and computer science conferences, such as ACM SIGMOD, VLDB, ICDE, PODS, ICDT, and ACM KDD. Potential topics include, but not are limited to: query languages, database system architectures, transaction management, data warehousing, XML and databases, data stream systems, wide scale data distribution, multimedia data management, data mining, and related subjects.

Probabilistic Databases
Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch
2011

Peer-to-Peer Data Management
Karl Aberer
2011

Probabilistic Ranking Techniques in Relational Databases
Ihab F. Ilyas and Mohamed A. Soliman
2011

Uncertain Schema Matching
Avigdor Gal
2011

Fundamentals of Object Databases: Object-Oriented and Object-Relational Design
Suzanne W. Dietrich and Susan D. Urban
2010

Advanced Metasearch Engine Technology
Weiyi Meng and Clement T. Yu
2010

Web Page Recommendation Models: Theory and Algorithms
Sule Gündüz-Ögüdücü
2010

Multidimensional Databases and Data Warehousing
Christian S. Jensen, Torben Bach Pedersen, and Christian Thomsen
2010

Database Replication
Bettina Kemme, Ricardo Jimenez Peris, and Marta Patino-Martinez
2010

Relational and XML Data Exchange
Marcelo Arenas, Pablo Barcelo, Leonid Libkin, and Filip Murlak
2010

User-Centered Data Management
Tiziana Catarci, Alan Dix, Stephen Kimani, and Giuseppe Santucci
2010

Data Stream Management
Lukasz Golab and M. Tamer Özsu
2010

Access Control in Data Management Systems
Elena Ferrari
2010

An Introduction to Duplicate Detection
Felix Naumann and Melanie Herschel
2010

Privacy-Preserving Data Publishing: An Overview
Raymond Chi-Wing Wong and Ada Wai-Chee Fu
2010

Keyword Search in Databases
Jeffrey Xu Yu, Lu Qin, and Lijun Chang
2009

Probabilistic Databases

Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch

ISBN: 978-3-031-00751-4 paperback
ISBN: 978-3-031-01879-4 ebook

DOI 10.1007/978-3-031-01879-4

A Publication in the Springer series
SYNTHESIS LECTURES ON DATA MANAGEMENT

Lecture #16
Series Editor: M. Tamer Özsu, *University of Waterloo*
Series ISSN
Synthesis Lectures on Data Management
Print 2153-5418 Electronic 2153-5426

Probabilistic Databases

Dan Suciu
University of Washington

Dan Olteanu
University of Oxford

Christopher Ré
University of Wisconsin-Madison

Christoph Koch
École Polytechnique Fédérale de Lausanne

SYNTHESIS LECTURES ON DATA MANAGEMENT #16

ABSTRACT

Probabilistic databases are databases where the value of some attributes or the presence of some records are uncertain and known only with some probability. Applications in many areas such as information extraction, RFID and scientific data management, data cleaning, data integration, and financial risk assessment produce large volumes of uncertain data, which are best modeled and processed by a probabilistic database.

This book presents the state of the art in representation formalisms and query processing techniques for probabilistic data. It starts by discussing the basic principles for representing large probabilistic databases, by decomposing them into tuple-independent tables, block-independent-disjoint tables, or U-databases. Then it discusses two classes of techniques for query evaluation on probabilistic databases. In *extensional query evaluation*, the entire probabilistic inference can be pushed into the database engine and, therefore, processed as effectively as the evaluation of standard SQL queries. The relational queries that can be evaluated this way are called safe queries. In *intensional query evaluation*, the probabilistic inference is performed over a propositional formula called *lineage expression*: every relational query can be evaluated this way, but the data complexity dramatically depends on the query being evaluated, and can be #P-hard. The book also discusses some advanced topics in probabilistic data management such as top-k query processing, sequential probabilistic databases, indexing and materialized views, and Monte Carlo databases.

KEYWORDS

query language, query evaluation, query plan, data complexity, probabilistic database, polynomial time, sharp p, incomplete data, uncertain information

Contents

Preface: A Great Promise

Traditional relational databases are *deterministic*. Every record stored in the database is meant to be present with *certainty*, and every field in that record has a precise, unambiguous value. The theoretical foundations and the intellectual roots of relational databases are in First Order Logic, which is essentially the relational calculus, the foundation of query languages such as SQL. In First Order Logic, the fundamental question is whether a logical sentence is true or false. Logical formulas under first-order semantics can be used to assert that a record is, or is not, in a relation, or in a query result, but they cannot make any less precise statement. The original applications that motivated the creation of relational databases required certain query results: accounting, inventory, airline reservations, and payroll. Database systems use a variety of tools and techniques to enforce this, such as integrity constraints and transactions.

Today, however, data management needs to include new data sources, where data are uncertain, and which are difficult or impossible to model with traditional semantics or to manage with a traditional Relational Database Management System (RDBMS). For an illustration, consider *Business Intelligence* (BI), whose goal is to extract and analyze business data by mining a large collection of databases. BI systems can be made more useful by including *external data*, such as twitter feeds or blogs, or email messages in order to extract even more valuable business information. For example, by analyzing blogs or twitter feeds and merging them with offline databases of products, companies can obtain early feedback about the quality of a new product or its degree of adoption, such as for a new car model, a new electronic gadget, or a new movie; such knowledge is very valuable, both for manufacturers and for investors. However, a traditional RDBMS requires the data to be precise: for each tweet, the system needs to know precisely what product it mentions and whether the comment is favorable or unfavorable. The data must be cleaned before it can be used in a traditional RDBMS.

The goal of *Probabilistic Databases* is to extend today's database technology to handle *uncertain* data. The uncertainty is expressed in terms of probabilities: a tuple is present only with some probability, or the value of an attribute is given by a probability distribution. Probabilistic databases are expected to scale as well as traditional database systems, and they should support queries as complex as those supported by advanced query processors today; however, but they will do this while allowing the data to be uncertain, or probabilistic. Both the data and their probabilities are stored in standard relations. The semantics, however, is *probabilistic*: the exact state of the entire database is not known with precision; instead, it is given by a probability distribution. When an SQL query is executed, the system returns a set of answers and it annotates each answer with a probability, representing the degree of confidence in that output. Typically, the answers are ranked in decreasing order of their output probability, so that users can inspect the top, most credible answers first. Thus, the main use of probabilities is to record the degree of uncertainty in the data and to rank the outputs to a

query; in some applications, the exact output probabilities matter less to the user than the ranking of the outputs. Probabilistic databases have a major advantage in processing uncertain data over their traditional counterparts. The data can be simply stored in the database without having to be cleaned first. Queries can be run immediately on the data. Cleaning can proceed gradually if and when more information becomes available by simply adjusting the probability value until it becomes 1.0, in which case the data becomes certain, or 0.0, in which case the data item can be removed. Even data that cannot be cleaned at all and will remain forever uncertain can still be stored and queried in a probabilistic database system.

Probabilistic databases take an evolutionary approach: the idea is to extend relational technology with a probabilistic semantics, rather than to develop a new artifact from scratch. All popular database techniques should carry over automatically to a probabilistic database: indexes, query optimization, advanced join algorithms, parallel query processing, etc. The goal is to extend the existing semantics of relational data to represent uncertainties but keep all the tools and techniques that have been proven so effective on deterministic data. As we will see in this book, this is not an easy task at all. The foundations of probabilistic databases are in First Order Logic *extended* with probabilities where the computational complexity of inference and model checking problems has only recently started to be understood.

The AI literature has studied probabilistic inference over *Graphical Models*, GM, such as Bayesian Networks and Markov Networks, which are described in several textbooks [Darwiche, 2009, Jordan, 1998, Koller and Friedman, 2009, Pearl, 1989]. There, the computational complexity is well understood: inference is exponential in the size of the network, and, to be more exact, in the *tree-width* of the network [Lauritzen and Spiegelhalter, 1990, Pearl, 1989]. Tree-width is a fundamental notion also in database theory: in particular, most interesting classes of queries require time exponential in the size of the query and, specifically, in the tree-width of its fundamental combinatorial structure, a graph [Abiteboul et al., 1995, Flum et al., 2002] or hypergraph [Chekuri and Rajaraman, 1997, Gottlob et al., 1999], formed by the relations occurring in the query as nodes and edges that represent joins. The difference here is that queries are usually small compared to the database, and query evaluation is easy under this assumption. By contrast, the network of a GM represents the data itself, and thus it can be very large.

The separation between query and data is a fundamental characteristics that distinguishes probabilistic databases from graphical models. The size of the data may be very large, but the queries are, by comparison, quite small. At a conceptual level, this distinction has been crisply articulated by Vardi [1982], who introduced the term *data complexity*. The query evaluation problem, both in traditional databases and in probabilistic databases, has two inputs: the query Q and the database instance D. In data complexity, the query Q is fixed, and the complexity is measured only as a function of the size of the database instance D. All modern query languages (SQL, XQuery) have polynomial time data complexity on deterministic databases[1], meaning that for any fixed Q, the

[1]The complexity of these languages becomes higher when extended with recursive functions, such as permitted in XQuery.

data complexity is in polynomial time in the size of the database. In contrast, there is no similar separation of query and data in GMs, where the entire network represents the data.

While it is possible to model a probabilistic database as a large graphical model, and reduce query evaluation to inference in GM [Sen and Deshpande, 2007], in this book we define and study probabilistic databases differently from GM. In our study, we separate the query from the data. We represent the uncertain data by a combination of classical database relations and propositional formulas, sometimes called *lineage*, which is an approach first introduced by Imieliński and Lipski [1984]. This approach leads us to probabilistic inference on propositional formulas, which, although being a special case of inference in GMs, has been investigated separately in the verification community by Bryant [1986] and in the AI literature by Darwiche [2009].

There are several reasons and advantages to this model of probabilistic databases over general GM.

First, under this model the database has a simple probabilistic model, which can scale easily. If a more complex probabilistic model is needed by the application, the correlations are expressed by the query (or view), which has a relatively small expression. This is a design principle that is well established in standard database modeling and schema normalization theory. In schema normalization, a deterministic table that has unwanted dependencies is decomposed into simpler tables that remove those dependencies and can be recovered from the decomposed tables using a view (usually a natural join). The same design principle exists in graphical models where a probability distribution on a large number of random variables is decomposed into a product of factors over smaller subsets of variables. The connection between database normalization and factor decomposition in graphical models was described by Verma and Pearl [1988]. Thus, in a probabilistic database, the base tables have a very simple probabilistic model, often consisting only of independent or disjoint tuples but can be very large, while the query may introduce complex correlations, but its expression is small. Independence properties in the data are in a strong sense certified by the representation and do not need to be discovered in the network structure of a GM by the inference algorithm. We explore representation formalisms for probabilistic databases in Chapter 2.

Second, the separation into query and data leads both to new inference techniques, specific to probabilistic databases, and to a better insight into the complexity of the probabilistic inference problem. We will describe a probabilistic inference method that is guided by the query expression and not by the database instance. In particular, one of the inference rules, the inclusion-exclusion formula or, more generally, the Möbius inversion function, has an exponential cost in the query yet a polynomial cost in the data: for that reason, inclusion-exclusion has no analog in traditional approaches for probabilistic inference on propositional formulas or in graphical models, yet, as we shall see, it proves to be very effective in probabilistic databases. The rule is possible only through the separation between the query and the data. At a theoretical level, the *data complexity* of probabilistic inference has an interesting dichotomy property: some queries Q have polynomial time data complexity, while others are provably hard for #P; every Union of Conjunctive Queries falls into one of these two categories; hence, the data complexity forms a dichotomy. This phenomenon does not have

a correspondence in other probabilistic inference settings since there is no distinction between the data and the query. We describe query evaluation on probabilistic databases in Chapter 3, Chapter 4, and Chapter 5.

Third, this query-centric approach to probabilistic inference allows us to build on decades of research on database management systems by reusing and extending database technology for data representation, storage, and query processing. It is a common theme in research on probabilistic database systems to build on existing database technology. Some of these approaches are surveyed in Chapter 6. Case studies of the TRIO and MayBMS systems can be found in the book by Aggarwal [2008].

This book contains a survey of the main concepts in probabilistic databases: representation formalisms for probabilistic data, query evaluation, and some advanced topics including sequential probabilistic databases, indexes, and Monte Carlo databases. Many applications today need to query large amounts of uncertain data, yet achieving scalability remains challenging. The techniques and concepts described in this book represent the state of the art in query processing on probabilistic databases. The new approach to probabilistic inference described in this book, based on the separation of the data and the query, holds a great promise for extending traditional, scalable database processing techniques with probabilistic inference. The book is intended for researchers, either in database or probabilistic inference, or as a textbook for an advanced graduate class.

Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch
May 2011

Acknowledgments

The authors would like to acknowledge many collaborators and friends who, through their discussions and comments have helped shape our thinking and, thus, have directly or indirectly influenced this book: Lyublena Antova, Magdalena Balazinska, Michael Benedikt, Nilesh Dalvi, Adnan Darwiche, Amol Deshpande, Daniel Deutch, Pedro Domingos, Robert Fink, Wolfgang Gatterbauer, Johannes Gehrke, Rainer Gemulla, Lise Getoor, Vibhav Gogate, Michaela Götz, Joseph Halpern, Andrew Hogue, Jiewen Huang, Thomas Jansen, Abhay Jha, Evgeny Kharlamov, Benny Kimelfeld, Phokion Kolaitis, Gerome Miklau, Tova Milo, Alexandra Meliou, Swaroop Rath, Karl Schnaitter, Pierre Senellart, Val Tannen, and Rasmus Wissmann.

Finally, the authors would like to acknowledge their funding agencies: Dan Suciu's work is supported by NSF IIS-0911036, IIS-0915054, IIS-0713576, and IIS-0627585. Dan Olteanu's work is supported by EPSRC under grant ADEPT number EP/I000194/1, and by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreements FOX number FP7-ICT-233599 and HiPerDNO number FP7-ICT-248135. Christopher Ré's work is supported by the Air Force Research Laboratory (AFRL) under prime contract no. FA8750-09-C-0181, the National Science Foundation under IIS-1054009, and gifts from Google, Microsoft, Physical Layer Systems, and Johnson Controls, Inc. Christoph Koch's work was supported by German Science Foundation (DFG) grant KO 3491/1-1, NSF grants IIS-0812272 and IIS-0911036, a KDD grant, a Google Research Award, and a gift from Intel. Any opinions, findings, conclusions, or recommendations expressed in this work do not necessarily reflect the views of DARPA, AFRL, or the US government.

Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch
May 2011