DBrev: Dreaming of a Database Revolution

Gjergji Kasneci
Microsoft Research
Cambridge, UK
gjergjik@microsoft.com

Jurgen Van Gael
Microsoft Research
Cambridge, UK
jvangael@microsoft.com

Thore Graepel
Microsoft Research
Cambridge, UK
thoreg@microsoft.com

ABSTRACT

The database community has provided excellent frameworks for efficient querying and online transaction or analytical processing. The main assumption underlying most of these frameworks is that there is no uncertainty regarding the stored data. However, in recent years, many important applications have emerged that need to manage noisy, corrupted, or incomplete data. This includes, e.g., anonymized data, data derived from sensor systems, or data from information extraction and integration systems. For such applications the assumption of logical consistency may not be valid and needs to be revised. In particular, techniques like probabilistic modelling and statistical inference may be necessary to be able to draw meaningful conclusions from the underlying data.

This paper presents DBrev, a hypothetical, intelligent database system for managing large quantities of data that involves uncertainty. We explain the main features of DBrev based on the scenario of information extraction and integration. We point out research challenges that need to be tackled and discuss a new set of assumptions that future database management frameworks need to build on.

General Terms

Databases, Probabilistic Reasoning, Information Extraction, Integration, Entity, Relationship, Knowledge Search

1. INTRODUCTION

For many decades the Database (DB) community has focused on applications involving data that is not subject to uncertainty or where the uncertainty can be ignored or managed outside of the database. Such applications include accounting, payroll, inventory, etc. However, for a wide variety of recently emerged applications, uncertainty is abundant and unavoidable: in many applications, measurement reading from sensors can be corrupted, noisy or involve missing data; in applications dealing with anonymized data, uncertainty is part of the ambiguity arising from missing values in the data; and most prominently, in information extraction and integration, uncertainty comes from the imperfect automatic extraction and disambiguation techniques or from unreliable sources. It is widely recognized by the DB community that the capabilities and the relational-algebraic models offered by state-of-the-art DB management systems are not sufficient for applications such as the above [16, 17, 10, 15, 19]. Rather, for such applications, there is a need for DB systems that can automatically quantify uncertainty, resolve inconsistencies, and provide means for ranked retrieval and knowledge discovery for the stored data based on uncertainty. The main problems that such a system should be able to deal with are:

Provenance The system needs to be able to reason about the derivation process and the validity of the stored data, as well as about the reliability of the data sources.

Context Awareness The system needs to keep track of the context in which data is valid. This may involve inferring entities and categories from the data, as well as reasoning about temporal, spatial and other relevant context.

Ambiguity The system needs to maintain different context-dependent interpretations of data and support the disambiguation process at query time. This may involve inferring probabilities over interpretations, depending on context, and possibly notions of statistically inferred semantic similarity.

Consistency The system needs to maintain consistency beyond logical integrity constraints. This includes more complex (first-order logic) inference rules on the one hand and the handling of soft, probabilistic constraints on the other.

Searching and Ranking The system needs to provide ranked retrieval and knowledge discovery mechanisms that can quickly adapt to the search context, preferences, and needs of the user.

The above problems have been addressed in isolation by different communities, e.g, Databases, Machine Learning, Information Retrieval, etc., and can be approached by current techniques. However, addressing and solving them simultaneously in an integrated system is, from our point of view, an extremely challenging (and hence “outrageous”) endeavor. The fundamental problem is to build the system on a framework for representing and updating beliefs under uncertainty. A promising candidate framework is probabilistic reasoning. Unfortunately, the scalable models used in state-of-the-art DB systems draw from first-order logic and are not designed to deal with probabilities. The Statistical Machine Learning (SML) community has given rise to comprehensive probabilistic reasoning models [20, 21, 22, 19], but these often still suffer from scalability issues. Jaynes' interpretation of probability theory as an extension of logic under uncertainty [21] points towards the commonalities of the DB and the SML communities. In this paper, we hypothetically join these two research avenues with the one of Information Retrieval, and present the hypothetical DBrev system as their synergetic yield. As an example, we explain how DBrev helps constructing and maintaining large-scale knowledge bases containing billions of entity-relationship-entity triples (statements) extracted from the Web and other sources. DBrev enables probabilistic reasoning and provides ranked retrieval and knowledge discovery over the stored knowledge. In order to mitigate the uncertainty inherent to information extraction and integration, DBrev aggregates statistics about different sources of evidence for the extracted triples, such as Web pages, extraction tools, Web 2.0 users, who may give feedback on the extracted triples, etc.

2. RELATED WORK

The main theoretical frameworks for combining the relational data representation with probabilistic reasoning are the *Probabilistic Database Model* and *Statistical Relational Learning*

Probabilistic Database Model (*PDM*) The PDM [16, 17, 10, 15] can be viewed as a generalization of the relational model which captures uncertainty with respect to the existence of database tuples (also known as tuple semantics) or to the values of database attributes (also known as attribute semantics). In the tuple semantics, the main assumption is that the existence of a tuple is independent of the existence of other tuples. Given a database consisting of a single table, the number of possible worlds (i.e. possible database instances) is 2^n, where n is the maximum number of the tuples in the table. Each possible world is associated with a probability which can be derived from the existence probabilities of the single tuples and from the independence assumption. In the attribute semantics, the existence of tuples is certain, whereas the values of attributes are uncertain. Again, the main assumption in this semantics is that the values attributes take are independent of each other. Each attribute is associated with a discrete probability distribution over the possible values it can take. Consequently, the attribute semantics is more expressive than the tuple-level semantics, since in general tuple-level uncertainty can be converted into attribute-level uncertainty by adding one more (Boolean) attribute. Both semantics could also be used in combination, however, the number of possible worlds would be much larger, and deriving complete probabilistic representations would be very costly. So far, there exists no formal semantics for continuous attribute values [16]. Another major disadvantage of PDMs is that they build on rigid and restrictive independence assumptions which cannot easily model correlations among tuples or attributes [10, 12, 19]. Such correlations, however, may be dictated by the application or domain at hand, and the underlying system has to provide a flexible framework to define and represent them.

Statistical Relational Learning (*SRL*) SRL models [12] are concerned with domains that exhibit uncertainty and relational structure. They combine a subset of relational calculus (first-order logic) with probabilistic graphical models, such as Bayesian or Markov networks to model uncertainty. These models can capture both, the tuple and the attribute semantics from the PDM and can represent correlations between relational tuples or attributes in a natural way [10]. More ambitious models in this realm are Markov Logic Networks [8, 19], Multi-Entity Bayesian Networks [13] and Probabilistic Relational Models [11]. Some of these models (e.g., [8, 13]) aim at exploiting the whole expressive power of first-order logic. While [8] represent the formalism of first-order logic by factor graphs, [11] and [13] deal with Bayesian networks applied to first-order logic. Usually, (approximate) inference in such models is performed using standard techniques such as belief propagation or Gibbs sampling. In order to avoid complex computations, [6, 7] propose the technique of lifted inference, which avoids materializing all objects in the domain by creating all possible groundings of the logical clauses. Although lifted inference can be more efficient than standard inference on these kinds of models, it is not clear whether they can be trivially lifted (see [9]). Hence, very often these models fall prey to high complexity when applied to practical cases. However, despite the complexity of probabilistic frameworks that build on graphical models, we think that future database systems can considerably benefit from lightweight graphical models for probabilistic reasoning, such as the ones presented in [14, 23].

Ranked Retrieval and Knowledge Discovery Finally, [4] and the references therein present approaches for combining Information Retrieval and Knowledge Discovery with current DB technology. Although the approaches discussed go a long way, they are rather static in nature by disregarding online updates of the data, which are inherent to many modern knowledge-oriented frameworks and applications, such as life-long information extraction [5], sensor networks and signal processing, etc. Most importantly, their frameworks do not consider holistic reasoning models for handling uncertainty.

3. DBREV

We illustrate the functionality of DBrev in the context of the management of information extracted from the Web. The system is continuously supplied with triples of the form <*entity*, *relationship*, *entity*>, where each triple comes with other metadata such as the URLs of Web pages from which it was extracted as well as temporal and/or spatial information about its validity (when available). In addition, DBrev continuously integrates implicit user feedback about the triples it contains; the feedback may be collected from an online game about encyclopedic knowledge or from users of Amazon's Mechanical Turk. The main tasks DBrev has to deal with are described in the following.

3.1 Data Provenance

The problem of data provenance (also known as the *lineage* problem) is closely related to the problem of database curation, which is an open problem in the presence of multiple information sources [16]. The idea is to trace the data derivation process back to the sources in order to guarantee data quality or to detect reasons for possible data inconsistencies. In probabilistic databases the lineage is handled by means of Boolean constraints on the tuples (e.g., *c-tables* [25]), which represent the set of possible worlds in which the tuples are true. In contrast, DBrev can compute the joint probability distribution over all possible worlds. Consequently, for any subset of triples, DBrev can return the maximum a posteriori assignment that maximizes their joint probability. Note that the triples can be related to each other through the sources they come from. Hence, DBrev constructs factor graphs in which the truth value of the triple is constrained by factors that relate it to variables quantifying the reliability of sources. This way the information sources become first-class citizens in DBrev. Furthermore, there can be other logical dependencies between the triples, such as dependencies concerning temporal and/or spatial dependencies [26]. These dependencies are translated into factor graphs as well, which are then integrated into the above factor graph. Consequently, they are handled as (soft) probabilistic constraints within the same reasoning framework (see Subsection 3.3). Efficient message passing on the factor graph corroborates the evidence and quantifies the uncertainty.

For example, consider the triple <*MichaelJackson*, *diedOn*, *25-07-2009*>. This triple could have been extracted from many different news pages and also from encyclopedic pages, such as Wikipedia. From a few other pages (e.g. `www.michaeljacksonsightings.com`), an extraction system could have extracted the triple <*MichaelJackson*, *seenIn*, *Cambridgeshire(UK)*> together with the temporal information '2010-03-08'[1]. In this case the corroboration process exploits temporal reasoning to decrease the truth value of the latter triple and the trust in `www.`

[1] In DBrev, temporal and spatial information about triples are represented by means of triple reification (see RDF Semantics at `http://www.w3.org/TR/rdf-mt/`).

michaeljacksonsightings.com. Note that the probabilistic corroboration problem is very subtle, as the truth values of triples and the trustworthiness of information sources are not necessarily determined by "majority voting" (e.g. by the number of Web pages or people who claim something, see also [23]). For example, if we corroborate user feedback about the triple <*BarackObama*, *hasWon*, *GrammyAward*> then a "majority voting" paradigm might fail since the majority of users may not know that Obama did indeed win the Grammy Award.

3.2 Ambiguity and Context Awareness

The ambiguity problem has been addressed in many variations, in different settings. It is one of the most acute problems in the field of information extraction and integration, where it arises in the form of *entity disambiguation/resolution*. For example, the integration of the datasets from different Social Web platforms, such as Facebook, MySpace, Twitter, flickr, LinkedIn, etc., poses a very hard problem, since the entities mentioned there can have ambiguous names. In the database setting, the problem occurs as the *record linkage* problem, where the goal is to find records that refer to the same entity. From a semantic point of view, the ambiguity problem is very difficult, as it often requires that contextual and background information be interpreted in the correct way. Hence, the ambiguity problem lies at the heart of AI. Consider the famous example sentence "The fruit flies like a banana". Contextual and background information play a decisive role for its understanding. At the same time, a probabilistic reasoning framework seems to be predestined for capturing the uncertainty that is inherent to disambiguation tasks.

For each entity, DBrev maintains two types of features: (1) ontological and (2) contextual features. While the contextual features are mainly provided by users and our extraction tools, the ontological features are automatically derived from general-purpose ontologies. For a given entity, the ontological features describe its taxonomic relations to other classes of entities (e.g. the entity *AlbertEinstein* belongs to the class *physicist*, *philosopher*, *person*, etc.). The contextual features consist of relevant terms (e.g. derived by frequency-based measures such as *tf-idf*) which occur in articles or user queries related to the given entity. While the ontological features represent some kind of commonsense background knowledge, the contextual features represent the different contexts that might be related to the given entity. The two types of features are combined into a unified representation and are used to map all the entities into a common latent space, in which the affinities or similarities between entities are measured. Similar ideas have been proposed in [24], where the authors describe a Bayesian model for the task of deriving feature-based similarities. On demand, the derived similarities allow DBrev to introduce for every pair of candidate entities e_1 and e_2 a new triple <e_1, *sameAs*, e_2>, which is assigned a corresponding probability (representing the belief that e_1 and e_2 are same) by the reasoning framework. This way, DBrev retains the flexibility to reassess its conclusions as new data comes in.

3.3 Consistency

In general, consistency can be viewed as a state (or possible world) in which a set of logical formulas are jointly satisfied. In databases, consistency is checked with respect to universal logical constraints (integrity constraints). A consistent transaction on a DB is one that does not violate those constraints. For example, the referential integrity constraints disallow dangling references, i.e., references to keys that do not exists in the DB.

DBrev exploits ontological knowledge, e.g. relationship properties, such as symmetry, transitivity, functionality[2], etc., to check whether the deductions between triples are consistent. Furthermore, as described in the previous subsection, DBrev combines the ontological knowledge with contextual knowledge to deal with ambiguity. The disambiguation component plays a critical role; without it the same entity might occur in the database in various dangling definitions, which would make logical deductions or transactions of any kind impossible. Consider the following rule, which describes the deduction of triples by exploiting the transitivity property of a relationship:

$$\begin{array}{l} <X,\ R,\ Y> \wedge <Y,\ R,\ Z> \wedge \\ <R,\ type,\ TransitiveRelation> \rightarrow <X,\ R,\ Z> \end{array}$$

where X, Y, and Z are entity variables, and R stands for a relationship variable. For example, from the triples <*MuséeDuLouvre*, *locatedIn*, *Paris*> and <*Paris*, *locatedIn*, *France*> DBrev can derive the triple <*MuséeDuLouvre*, *locatedIn*, *France*>. Although the latter triple may not be explicitly stored in the database, its derivation is very useful for the reasoning process, since it represents a logical constraint between triples. This is exploited to support the lineage (see Subsection 3.1) and the disambiguation (see Subsection 3.2). Consider a newly extracted triple <*"Louvre"*, *"is located in"*, *"France"*>. DBrev supports its disambiguation component by reasoning probabilistically about logical rules of the following kind:

$$\begin{array}{l} refersTo(\text{“}r\text{”},\ R) \wedge \\ refersTo(\text{“}y\text{”},\ Y) \wedge \\ canBeDeduced(X,\ R,\ Y) \wedge D \\ \rightarrow refersTo(\text{“}x\text{”},\ X) \end{array}$$

where D represents a conjunction of contextual constraints (e.g., temporal, spatial, or domain-based constraints), R represents a relationship variable, and X and Y represent entity variables. This way DBrev can become more confident in the hypothesis that "Louvre" is a useful description for the entity *MuséeDuLouvre*. Similar rules were introduced in [1] to support the disambiguation process. However, DBrev allows users to define a wide range of logical constraints, which are interpreted as probabilistic rules (i.e., soft constraints) on the stored data; [23] shows how similar deduction rules can be translated into factor graphs.

3.4 Searching and Ranking

For large-scale information retrieval tasks such as web search, the ranking-oblivious conditions of Boolean search, which were mainly used for querying library or product catalogs, have been replaced by similarity and preference based ranking techniques involving vectorial or bag-of-words representations of documents and queries. Following the same trend, DBrev combines the unstructured conditions of keyword retrieval and the structured query paradigm of databases with question answering techniques, while making ranking a first-class citizen. This allows casual as well as expert users to query the system. The search and ranking model of DBrev is based on the following desiderata:

Pattern-Based Approximate Matching DBrev is geared to answer knowledge queries (i.e., queries that ask about entities and relationships between them) or questions. Knowledge queries can be expressed through a graph-based query language similar to the one proposed in [3]. An example search task could be: *"Find all US companies that are certified partners of Microsoft"*. Figure 1 depicts a graph-based representation of this query. The node labeled with \$x represents a variable, which in the answering phase is replaced by entities that satisfy the relationship constraints given by the query graph. The expression *locatedIn** aims at capturing geographical hierarchies, e.g. cities, counties, states, countries, etc. Furthermore, node and edge labels are relaxed through labels that refer to the same entities and relations, respectively. For example, the node labeled "Microsoft" is relaxed through labels that might refer to the same real-world entity, e.g., "MS", "MS Corporation",

[2] E.g. the relationship *X born on date Y* is functional, since every person can only have one date of birth.

"MSFT", etc. The entity disambiguation component of DBrev takes care of retrieving similar labels for relaxation. Natural language questions are first translated to graph-based queries, which are then answered by means of the same relaxation technique.

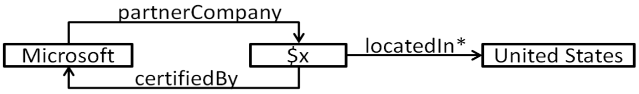

Figure 1: Graph-based representation of the search task "Find all US companies that are certified partners of Microsoft"

Top-k Ranking for Multiple Criteria Queries such as the above have a knowledge discovery character and may return a result set that is too large for a human to handle. This means that the results need to be ranked with respect to various criteria. The main criteria in DBrev are (1) similarity and (2) user preference. In an approximate matching paradigm, ranking by similarity (e.g. entity- or relationship-based similarity) is crucial. This allows DBrev to rank salient results higher than results that may be only vaguely related to the query. However, from a user perspective, the ranking becomes really meaningful if the system takes the user preferences into account. This is why DBrev makes use of the user context (e.g., location, background, general and current interests, etc.) and takes into account his information needs (e.g. information freshness, accuracy, popularity, etc.). Since the above criteria involve probabilities, which need to be aggregated in an efficient way, DBrev computes the results in a top-k fashion. This in lines with [18], where Ré et al. argue that in a probabilistic setting, the only meaningful semantics for returning results to a user is by ranking them. Finally, DBrev allows users to specify their own ranking criteria and provides hyperbolic visualization tools for data exploration.

4. CONCLUSION

In this "outrageous" paper we have speculated about a direction towards which database research may evolve. Our dream database system DBrev combines ideas from database research, machine learning and information retrieval to be able to manage the huge amounts of unreliable information extracted from the web. The challenge of large-scale information extraction illustrates how we need to employ and extend the notions of provenance, context, ambiguity, consistency, and ranking as key concepts for future database research. Although we have circumvented many other important questions (e.g. dynamic index updates, multidimensional indexing, etc.), we hope that the above mentioned research communities may take some inspiration from our dream and may seize the opportunity to collaborate on the challenges ahead.

5. REFERENCES

[1] Suchanek, F. M., Sozio, M., Weikum, G.: SOFIE: Self-Organizing Flexible Information Extraction. In: 18th International World Wide Web conference (WWW 2009), pp. 631–640. ACM Press (2009)

[2] Suchanek, F. M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: 16th International World Wide Web Conference (WWW 2007), pp. 697–706. ACM Press (2007)

[3] Kasneci, G., Suchanek, F. M., Ifrim, G., Ramanath, M., Weikum, G.: NAGA: Searching and Ranking Knowledge. In: 24th International Conference on Data Engineering (ICDE 2008), pp. 953–962. IEEE (2008)

[4] Weikum, G., Kasneci, G., Ramanath, M., Suchanek, F.: Database and Information-Retrieval Methods for Knowledge Discovery. In.: Communications of the ACM (CACM 2009), pp. 56–64. ACM Press (2009)

[5] Banko, M., Etzioni, O.: Strategies for Lifelong Knowledge Extraction from the Web. In: 4th International Conference on Knowledge Capture (K-CAP 2007), pp. 95–102. ACM Press (2007)

[6] Poole, D.: First-Order Probabilistic Inference. In: 8th International Joint Conference on Artificial Intelligence (IJCAI 2003), pp. 985–991, Morgan Kaufmann (2003)

[7] Domingos, P., Singla, P.: Lifted First-Order Belief Propagation. In: 23rd AAAI Conference on Artificial Intelligence (AAAI 2008), pp. 1094–1099. AAAI Press (2008)

[8] Domingos, P., Richardson, M.: Markov Logic Networks. In: Machine Learning, 62(1–2), pp. 107–136. Springer (2006)

[9] Jaimovich, A., Meshi, O., Friedman, N.: Template Based Inference in Symmetric Relational Markov Random Fields. In: 23rd Conference on Uncertainty in Artificial Intelligence (UAI 2007), pp. 191-199. AUAI Press (2007)

[10] Sen, P., Deshpande, A., Getoor, L.: PrDB: Managing and Exploiting Rich Correlations in Probabilistic Databases. In: Journal of Very Large Databases, 18(5), pp. 1065–1090. Springer (2009)

[11] Friedman, N., Getoor, L., Koller, D., Pfeffer, A. Learning Probabilistic Relational Models. In: 16th International Joint Conference on Artificial Intelligence (IJCAI 1999), pp. 1300–1309. Morgan Kaufman (1999)

[12] Getoor, L.: Tutorial on Statistical Relational Learning. In: 15th International Inductive Logic Programming Conference (ILP 2005), Springer (2005)

[13] Da Costa, P. C. G., Ladeira, M., Carvalho, R. N., Laskey, K. B., Santos, L. L., Matsumoto, S.: A First-Order Bayesian Tool for Probabilistic Ontologies. In: 21st International Florida Artificial Intelligence Research Society Conference (FLAIRS 2008), pp. 631–636. AAAI Press (2008)

[14] Wang, D. Z., Michelakis, E., Garofalakis, M., Hellerstein, J. M.: BayesStore: managing large, uncertain data repositories with probabilistic graphical models. In: 34th International Conference on Very Large Data Bases (VLDB 2008), 1(1), pp. 340–351, ACM Press (2008)

[15] Antova, L., Koch, C., Olteanu, D.: 10^{10^6} Worlds and Beyond: Efficient Representation and Processing of Incomplete Information. In: 23rd International Conference on Data Engineering (ICDE 2007), pp. 606–615. IEEE (2007)

[16] Dalvi, N. N., Ré, C., Suciu, D.: Probabilistic Databases: Diamonds in the Dirt. In: Communications of ACM, 52(7), (CACM 2009), pp. 86–94. ACM Press (2009)

[17] Agrawal, P., Benjelloun, O., Sarma, A. D., Hayworth, C., Nabar, S. U., Sugihara, T., Widom, J.: Trio: A System for Data, Uncertainty, and Lineage. In: 32nd International Conference on Very Large Data Bases (VLDB 2006), pp. 1151–1154. ACM Press (2006)

[18] Ré, C., Dalvi, N., Suciu, D.: Efficient Top-k Query Evaluation on a Probabilistic Database. In: 23rd Very Large Databases Conference (VLDB 2007), pp. 51–62. ACM Press (2007)

[19] Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning. MIT press (2007)

[20] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1997)

[21] Jaynes, E. T.: Probability Theory – The Logic of Science. Cambridge University Press (2003)

[22] Bishop, C. M.: Pattern Recognition and Machine Learning. Springer (2007)

[23] Kasneci, G., Gael, J. V., Herbrich, R., Graepel, T.: Bayesian Knowledge Corroboration with Logical Rules and User Feedback. In: ECML PKDD 2010, Springer (2010)

[24] Stern, D., Herbrich, R., Graepel, T.: Matchbox: Large Scale Bayesian Recommendations. In: International World Wide Web Conference WWW 2009 (2009), pp. 111–120, ACM Press (2009)

[25] Imielinski, T., Lipski, W.: Incomplete Information in Relational Databases. In: Journal of the ACM, 31, pp. 761–791, ACM Press (1984)

[26] Gérard Ligozat, G., Mitra, D., Condotta, J.: Spatial and temporal reasoning: beyond Allen's calculus. In: AI Communications, 17(4), pp. 223-233, IOS Press (2004)