

Probabilistic Data Integration Systems

Dan Suciu*

Current data integration techniques are successful at managing well-defined and well-understood data integration tasks, but do not cope well with uncertainty. However, the amount of uncertain data is growing with the number and variety of data sources being integrated, both in traditional data integration tasks s.a. enterprise data integration, and in next generation integration problems, s.a. combining structured data with information extracted automatically from text.

We argue that uncertainty in data integration is best coped with by using a probabilistic data model, and propose here *probabilistic data integration systems*, PDIS. Our justification comes from the fact that probabilities can be used to model a rich variety of imprecisions arising in information integration. For example:

- Every information integration system needs to run some kind of object reconciliation algorithm to match similar objects from two different sources. For example, because the representation of gene differs slightly between Entrez and Swissprot, any integration system must match genes from the two sources. Algorithms for computing definitive matches are domain specific, expensive, and most often imperfect. By contrast, a PDIS allows *probabilistic matches*, in which a probabilistic score is assigned to each pair of potential matches. This lowers the cost of object reconciliation, allowing information integration to scale to a larger number of data sources. Moreover, while in some domains, such as people, companies, or product names, a golden standard for object matching usually exists, in others no such standard exists. For example, when using Blast to match two sequences from two data sources, there is no golden standard for coercing its output score into a categorical match or non-match, and one must use the resulting match as probabilistic.
- In automatic information extraction (IE) from text, each extracted data item has an associated confidence score that indicates the system's confidence. Current IE systems compute the scores, but then ignore them, or simply return them to the user. In contrast, a PDIS can handle *probabilistic data* natively, thus further using these scores when integrating the extracted data with information from other sources, such as relational databases, or even another IE sources.
- Despite extensive research on schema matching techniques, fully automatic schema matching between large database schemata is not possible. In contrast, a PDIS accepts *probabilistic schema matches*, e.g. consisting of pairs of potentially matching schema

*University of Washington

items together with a confidence score. A PDIS therefore significantly reduces the cost of schema integration by allowing it to be fully automated and thus scale to large number of data sources.

- Scientific data often has some explicit information indicating the author’s degree of confidence in that data. Data uncertainties arise from factors such as the amount of additional supporting data, the experimental methods and controls employed, as well as the degree and type of processing undergone by the data. Such confidence information is usually available in scientific data, and a PDIS can take it into account when integrating it with other data sources.

The science of PDIS is grounded in both traditional data integration and in probabilistic inference techniques in knowledge representation. Yet they face major new challenges.

One major challenge is scalability and query performance. To scale to large amounts of data PDIS need to combine both traditional query processing techniques (access methods, indexes, cost based optimizations) and probabilistic inference methods. It is known that, when evaluated over probabilistic data, SQL queries have high complexity (#P-complete). To allow efficient query processing, an optimizer needs to carefully isolate the part of the query that can be pushed in a database engine from that which requires an expensive probabilistic inference. Another issue is that a PDIS needs a new query answering paradigm that allows users to retrieve and explore the top k answers much easier than the rest. Since probabilities in a PDIS model imprecisions, the answer to a typical query contains too many noisy items (false positives). Rather than computing all answers (which is expensive) the system should focus on retrieving and ranking the top k. New techniques, ranging from query processing and optimization to data visualization, are required to support the new top-k answering paradigm in PDIS.

Another major challenge is the need for tools that allow users to gradually clean and improve the quality of the probabilistic integrated data. We argue here that *probabilistic constraints* represent a very promising starting point for designing such tools. Traditional database constraints are routinely used today in relational databases, allowing users to enrich the semantics of their data by declaring invariants over the data. In a PDIS, however, a probabilistic constraint can be used to clean and to enhance the quality of the data. A PDIS needs to support both hard and soft constraints. For example *each company name must match exactly one name from a reference list of companies* is a *hard* constraint, and impacts the probability distribution on all possible matches between a set of company names and a reference set of companies. By contrast, *with confidence 80% all company names extracted from a corpora of text also appear in a reference list of companies* is a *soft* constraint, which affects the probabilities of some extracted data. Probabilistic constraints can be classified by the way they are derived: they can be *stated* by the administrator, or they can be *learned* automatically from some training data. Also, probabilistic constraints can be used in multiple ways: they can be used to *clean* existing data (e.g. by altering probabilities of the data) or they can be used in some sort of *semantic query optimizations*.

Coping with uncertainty and imprecisions is critical to future data integration systems, hence we advocate an intensive study of various aspects of probabilistic data integration systems. Yet the scientific challenges are major, and will require years of research from several communities: databases, knowledge representation, machine learning, and algorithms.