Elicitation, Estimation & Explanation Challenges in Handling Imprecision & Incompleteness in Autonomous Databases

(Position Paper for Penn II Workshop)

Topic: Imprecision and uncertainty in data and inferences

Subbarao Kambhampati*
Department of Computer Science and Engineering,
Arizona State University, Tempe, AZ 85287, USA

ABSTRACT

We will motivate the problem of simultaneously handling incompleteness and imprecision in autonomous databases. We will argue that effectively tackling this problem requires solutions to density and relevance estimation, query rewriting and result explanation. We will show that solving these problems requires tools from decision theory, utility elicitation, statistical learning, as well as core database techniques for handling uncertainty and incompleteness. We will provide pointers to our current progress in designing a system, QUIC, for handling some of these challenges.

Motivation

The popularity of the World Wide Web has lead to the presence of multiple online databases that are web-accessible to lay users. An important challenge for the database community is to develop mediators that can enable lay users to seamlessly accessing these databases. Two of the challenges faced by such mediators involve handling incompleteness and imprecision:

Incompleteness: An increasing number of online databases are being populated with little or no curation (either through automated extraction or by lay users). These methods, which are often used to populate many of today's web databases, are error prone. This leads to databases that are *incomplete* in that they may contain tuples having many null values. For example, databases such as `autotrader.com` which are populated by crawling text classifieds and by car owners entering the data through forms. On a random sample of 25,000 tuples of `autotrader.com`, we found that almost 34% were incomplete! This type of incompleteness is expected to increase even more with services such as GoogleBase which provide users significant freedom in deciding which attributes to define and/or list.

Imprecision: More and more lay users are accessing databases on the web. Users often are not clear about their needs and the query formulating tools provided such as forms do not support imprecision. Hence users end up asking queries such as $Q : CarDB(Model = Civic)$ when they actually want to ask the query $Q' : CarDB(Model \approx Civic)$ (where "≈" is to be interpreted as "like"). A mediator system should be able to support the queries of the later kind.

In both cases, there are some tuples which do not exactly satisfy the query constraints but nevertheless are likely to be relevant to the user. This makes query processing on databases face challenges that traditionally had to be handled only by information retrieval systems.

Example: Consider a fragment of online Car database DB_f as shown in Table 1. Suppose a user poses an imprecise query $Q' : \sigma_{Model \approx Civic}$ on this table. Clearly tuples t_1 and t_6 are relevant as they are exact answers to the query. In addition, the tuples t_4 and t_7 might also be relevant if there are reasons to believe that the missing value might be a Civic. Finally, tuples t_2 and t_8 might be relevant if Civic is considered similar enough to Accord and/or Prelude.

Id	Make	Model	Year	Color	Body Style
1	Honda	Civic	2000	red	coupe
2	Honda	Accord	2004	blue	coupe
3	Toyota	Camry	2001	silver	sedan
4	Honda	null	2004	black	coupe
5	BMW	3-series	2001	blue	convt
6	Honda	Civic	2004	green	sedan
7	Honda	null	2000	white	sedan
8	Honda	Prelude	1999	blue	coupe

Table 1: Fragment of a Car Database

Our Approach

Ideally, a query processor should be able to present such implicitly relevant results to the user, ranking them in terms of their expected relevance. We are working towards a general framework as well as a specific current implementation, called QUIC[1], aimed at this problem. We start with the philosophy that imprecision and incompleteness are best modelled in terms of relevance and density functions. Informally, the relevance function assesses the value the user places on a tuple t with respect to a query Q. The density function attempts to capture the probability distribution that an incomplete tuple $\widehat{t}$ is in reality representing a complete tuple

*This is joint work with Yi Chen, Jianchun Fan, Hemal Khatri, Ullas Nambiar and Garrett Wolf. A longer version of the paper with additional technical details of the QUIC approach is available at *rakaposhi.eas.asu.edu/quic-short.pdf*. Additional papers on information integration from our group are available at *rakaposhi.eas.asu.edu/i3*

[1]QUIC is an acronym for Querying(Q) under(U) Imprecision(I) and Uncertainty(C)

t. Table 2 gives examples of density and relevance functions for our car scenario.Given such information about relevance and density functions, it is possible, in theory, to rank a set of possibly incomplete tuples in terms of their expected relevance given a specific query.

$\sigma_{Model \approx Civic}$	*Civic*	*Accord*	*Prelude*	*Corolla*
Relevance	1.0	0.78	0.59	0.48
Density	0.62	0.21	0.17	0.0

Table 2: Relevance for the query Q': $\sigma_{Model \approx Civic}$ and Density for tuple $\widehat{t_4}$.

Challenges

Simple as it sounds, realizing such a query processing architecture presents several technical challenges, brought about mostly due to the autonomous nature of the databases, and the impatience of the user population.

1. **Ranking:** What is the best way to combine the density and relevance functions to provide a ranking of the tuples? Ideally, the ranking method should be "downward compatible"–in that it should also work in scenarios that do not have incompleteness and/or imprecision.
2. **Assessment of Relevance function:** Since the users are often impatient, we need methods for automatically and non-intrusively assessing the appropriate relevance function.
3. **Assessment of Density function:** Since the databases are autonomous, we need methods for automatically assessing the density function.
4. **Retrieving Relevant tuples:** Since often the query processor has only a form-based interface to the databases, we need to be able to rewrite the queries appropriately in order to retrieve tuples that are likely to be relevant to the user.
5. **Explaining/Justifying Answers:** Since the query processor will be presenting tuples that do not correspond to exact answers to the user's query, it needs to provide explanations and justifications to gain the user's trust.

Novel Aspects over related problems

The problem we address brings together several core database challenges — including uncertainty, incompleteness and imprecision, to the context of web with its autonomous databases and lay user population. The problems of incompleteness and imprecision have, in the past, been tackled in isolation in traditional databases, but the methods tended to involve direct modification of the databases, to rewrite null values (e.g. [6]) or to insert similarity measures between attribute values (e.g. [9]). The autonomous nature of the web sources, as well as the impatience of the web users craving instant gratification, precludes straightforward adaptation of these methods. The complications include the need to automatically, indirectly and non-intrusively assess relevance and density functions, as well as the need to use these assessed functions in reformulating the user query so as to retrieve tuples that are expected to be of relevance to the user. Assessment of relevance and density functions are special cases respectively of the general problems of "utility/preference elicitation"[3, 4] and "density estimation"[5] that have been studied extensively in the AI, statistics and pattern recognition communities. The challenge we face is adapting, from this welter of literature, the right methods that are sensitive to the autonomous nature of the databases as well as the possibly non-cooperative user population.

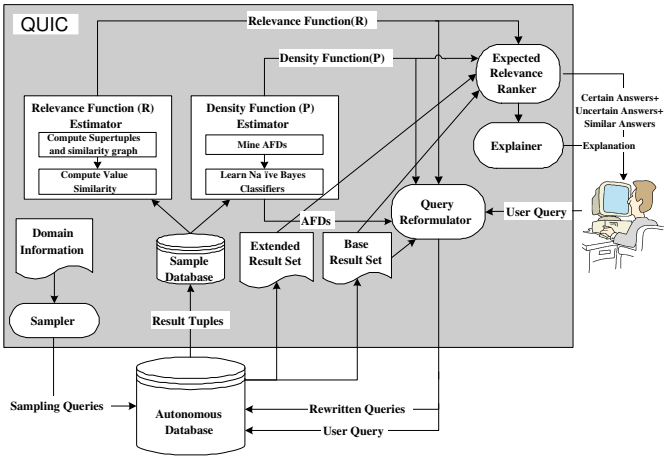

Figure 1: QUIC System Architecture.

Given that our treatment of incomplete tuples involves converting them into uncertain tuples, work on probabilistic databases[2, 1] shares some of our goals. It however focuses on single centralized databases and assumes that the uncertainty in the base data is already quantified. In contrast, we focus on autonomous databases, and thus must assess the uncertainty in the base data automatically.

Current Progress

Our work on the AIMQ system [10] focuses exclusively on imprecision, while our recent work on QPIAD [8] focuses exclusively on incomplete information. Our most recent work on QUIC [7] extends over those prior systems both in terms of addressing imprecision and incompleteness together, as well as investigating a larger spectrum of methods for assessing relevance and density functions, and using them for query rewriting. The current architecture of QUIC is shown in Figure 1. QUIC acts as a mediator between the user and autonomous web databases. Because of the autonomous nature of these databases, QUIC has virtually no control or prior knowledge over them. QUIC computes relevance and density functions from the probed samples of the autonomous databases. These functions are in turn used to reformulate the user query in order to retrieve tuples that are relevant even though they may not be exact answers. The retrieved tuples are ranked in terms of a ranking function that we call *expected relevance ranking*. The ranked results are returned with an explanation of the ranking scheme. Rather than assess user-specific relevance functions, QUIC attempts to assess relevance functions for the whole user population, based on value similarity and attribute importance. For density estimation, the current prototype assumes independence between attributes, and assesses the density function by learning an AFD-enhanced Naïve Bayes Classifier(NBC) from a sample database [8]. Additional details of the approach are provided in [7]. At the time of this writing, we are focusing on three important outstanding challenges: (i) handling attribute correlations in assessing relevance and density functions (ii) supporting more expressive queries, including joins and (iii) focusing on data-integration issues that arise in the context of multiple autonomous databases.

1. REFERENCES

[1] P. Andritsos, A. Fuxman and R.J. Miller. Clean Answers over Dirty Databases: A Probabilistic Approach. ICDE 2006: 30

[2] O. Benjelloun, A. Das Sarma, A. Halevy, and J. Widom. ULDBs: Databases with Uncertainty and Lineage. VLDB 2006.

[3] J. Blythe. Visual exploration and incremental utility elicitation. In *AAAI*, 2002.

[4] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research(JAIR)*, 2003.

[5] T. Hastie, R. Tibshirani and J. Friedman. Elements of Statistical Learning. Springer Verlag (2001).

[6] T. Imielinski, and W. Lipski Jr. Incomplete Information in Relational Databases. In *J. ACM 31(4): 761-791*, 1984.

[7] Subbarao Kambhampati, Yi Chen, Jianchun Fan, Hemal Khatri, Ullas Nambiar and Garrett Wolf. Handling Imprecision & Incompleteness in Autonomous Databases. ASU CSE TR 06-014. July 2006.

[8] H. Khatri, J. Fan, Y. Chen, and S. Kambhampati. Query processing over incomplete autonomous databases. *ASU CSE TR-06-006, rakaposhi.eas.asu.edu/qpiad-tr.pdf*, 2006.

[9] A. Motro. Vague: A user interface to relational databases that permits vague queries. ACM TOIS 6(3), 1998.

[10] U. Nambiar and S. Kambhampati. Answering imprecise queries over autonomous web databases. In *ICDE*, 2006.

[11] R.E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall. 2004.