QPIAD: Query Processing over Incomplete Autonomous Databases

Hemal Khatri Jianchun Fan Yi Chen Subbarao Kambhampati
Department of Computer Science and Engineering
Arizona State University
{hemal.khatri,jcf,yi,rao}@asu.edu

Abstract

Incompleteness due to missing attribute values (aka "null values") is very common in autonomous web databases, on which user accesses are usually supported through mediators. Traditional query processing techniques that focus on the strict soundness of answer tuples often ignore tuples with critical missing attributes, even if they wind up being relevant to a user query. Ideally we would like the mediator to retrieve such relevant uncertain answers and gauge their relevance by accessing their likelihood of being relevant answers to the query. However, the autonomous nature of the databases poses several challenges, such as the restricted access privileges, limited query patterns, and sensitivity of database and network resource consumption in the web environment. We introduce a novel query rewriting and optimization framework QPIAD that tackles these challenges to retrieve relevant uncertain answers. Our technique involves reformulating the user query based on approximate functional dependencies (AFDs) among the database attributes and ranking these queries using value distributions learned from Naïve Bayes Classifiers. Empirical studies demonstrate the effectiveness of our approach in retrieving relevant uncertain answers with high precision, high recall and manageable cost.

Categories and Subject Descriptors: Managing uncertain data, mediator query processing
Keywords: query rewriting, incomplete databases, autonomous databases, querying hidden web

1 Introduction

Data integration in autonomous web databases has drawn much attention in recent years, as more and more data in the back-end databases becomes accessible via web servers. A mediator provides a unified query interface as a global schema of the underlying databases. Queries on the global schema are then rewritten as queries over autonomous databases through their web interfaces. Current mediator systems [5, 3] return to user only *certain answers* that exactly satisfy all the user query predicates. Tuples that are otherwise highly relevant for the query will not be retrieved if they have null values on any of the query predicates. For example, in a used car trading application, if a user asks for convertible cars, all the returned answers must have the value "convt" for the attribute $body\ style$. Even though all Z4's are convertibles, a BMW Z4 car which has a null value in its $body\ style$ will not be returned. This is particularly problematic when the data sources have a significant fraction of incomplete tuples, and/or the user requires high recall (consider, for example, a law-enforcement scenario, where a potential crime suspect goes unidentified because of information that is fortuitously missing in the database).

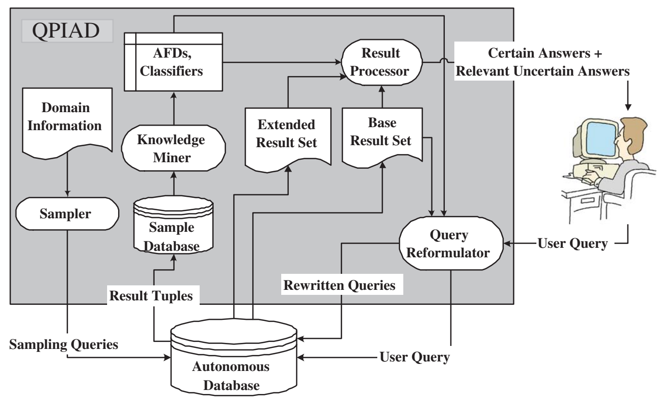
Incompleteness can creep into online databases for many reasons the primary one being incomplete entry by lay individuals,[1] inherent inaccuracy of automated extraction techniques and schema heterogeneity between local and global schemas. A tuple $t \in$ Relation R is said to be complete if it has non-null values for each of the attributes A_i; otherwise it is considered incomplete. A random sample of 25,000 tuples extracted from two popular car trading websites had 33% and 98% tuples containing missing values!

When faced with such incomplete databases, current mediators that provide only *certain answers* thereby sacrifice recall. A naïve approach for improving the recall would be to return, in addition to the certain answers, all the tuples with missing values on the constrained attribute(s), referred to as *uncertain answers*. This approach of returning *All Possible Answers* referred to as ALLPOS has two obvious drawbacks. First, it is infeasible to retrieve all the tuples with nulls due to limited query access patterns supported by web databases. Second, and perhaps more important, many tuples with missing values on constrained attributes are *irrelevant* to the query. The ALLPOSapproach thus improves recall but suffers from low precision and high cost.

In this paper, we present *QPIAD*, a system that supports **q**uery **p**rocessing over **i**ncomplete **a**utonomous **d**atabases, which not only returns certain answers but also returns, in a ranked fashion, tuples that have missing values and yet are highly relevant to queries.

The *QPIAD* system architecture is shown in Figure 1. In this framework, a user accesses autonomous databases through a mediator. When a user submits a query to the mediator, the query reformulator first directs the query to the autonomous databases and retrieves the set of all certain answers (called the *base result set*). In order to retrieve relevant uncertain answers, the mediator needs to issue additional queries taking into account the limited access pat-

[1]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.



ID	Make	Model	Year	Body style
1	Audi	A4	2001	convt
2	BMW	Z4	2002	convt
3	Porsche	Boxster	2005	convt
4	BMW	Z4	2003	null
5	Honda	Civic	2004	null
6	Toyota	Camry	2002	sedan

terns of the autonomous databases. We propose online query rewriting techniques to generate new queries based on the original query, the base result set, and attribute correlations learned from a database sample. The attribute correlations used to generate the rewritten queries are mined in terms of Approximate Functional Dependencies (AFDs)[2]. The goal of these new queries is to return an *extended result set*, which consists of highly relevant uncertain answers to the original query. Since these rewritten queries are not all equally good in terms of retrieving relevant uncertain answers, they are ranked before being posed to the databases. The ranking of rewritten queries is based on the value distributions for the missing attribute. We reduce the problem of acquiring such value distributions to learning classifiers, and develop an AFD-enhanced Naïve Bayesian classifier learning method (where AFD plays a feature selection role for the classification task). As shown in Figure 1, *QPIAD* mines attribute correlations and learns value distributions on a small portion of data sampled from the autonomous databases. The sampling module collects the sample data from the autonomous databases using random probing queries, and the knowledge mining module learns AFDs and the AFD-enhanced classifiers from these samples.

Experimental evaluation shows that *QPIAD* is effective in retrieving query answers from autonomous incomplete databases with good precision, recall and manageable cost.

2 Retrieving Relevant Uncertain Answers from Autonomous Databases

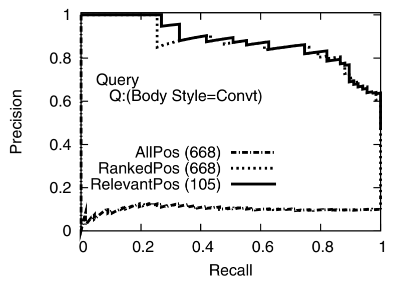
Query Rewriting: The goal of our query rewriting is to generate a set of rewritten queries to retrieve relevant uncertain answers. Consider a selection query Q for cars having $body\ style{=}convt$ on a fragment of a Car database as shown in Table 1. To process user query Q on data in Table 1, we first issue Q on the autonomous database to retrieve all the certain answers which consist of tuples t_1, t_2 and t_3. These certain answers form the *base result set* of Q. Consider $t_1{=}\langle Audi, A4, 2001, convt\rangle$, if there is a tuple t_i in the database with the same value for $model$ as t_1 but missing value for $body\ style$, then $t_i.body\ style$ is likely to be $convt$. We *capture this intuition by mining attribute correlations* from the data itself.

One obvious type of attribute correlations is "*functional dependencies*". For example, the functional dependency $model{\rightarrow}make$ often holds in automobile data records. However, often there are not enough functional dependencies in the data, and autonomous databases are unlikely to advertise the functional dependencies. The answer to both these problems involves *learning* approximate functional dependencies from a (probed) sample of the database. $X{\rightsquigarrow}A$ over relation R is an *approximate functional dependency* (AFD) if it holds on all but a small fraction of the tuples. The set of attributes X is called the *determining set* of A denoted by dtrSet(A).

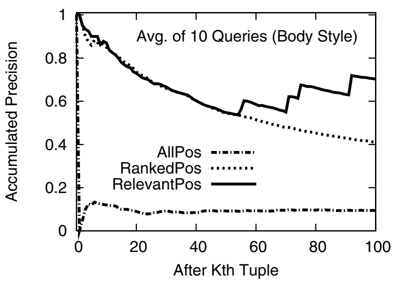
For example, consider an AFD $model{\rightsquigarrow}body\ style$ in the sample Car database, which indicates that the value of a car's $model$ attribute *sometimes* (but not always) determines the value of $body\ style$ attribute. Therefore, we consider t_i as a relevant uncertain answer to the query Q. To retrieve t_i from the database (which does not support binding on null values), the mediator can issue another query $Q_1{:}\,\sigma_{model=A4}$ based on the *determining set* of the attribute $body\ style$. Similarly, we can issue queries $Q_2{:}\,\sigma_{model=Z4}$ and $Q_3{:}\,\sigma_{model=Boxster}$ to retrieve other relevant uncertain answers. Note that the generated queries will return highly relevant uncertain answers, such as t_4, as well as a few tuples whose $body\ style$ value is neither $convt$ nor $null$, as the AFD only holds approximately. Therefore, we filter out those tuples.

Since the proposed approach aims to restrict the retrieval of uncertain answers to just the *relevant* tuples, it is named as returning *Relevant Possible Answers(*RELEVANTPOS*)*. RELEVANTPOS approach has two advantages. First, it can be used to query autonomous databases which do not support null value binding. Second, RELEVANTPOSis much more efficient as it only retrieves relevant uncertain answers rather than all uncertain answers thus requiring fewer tuples to be retrieved and transmitted.

Ranking Rewritten Queries: In the query rewriting step of RELEVANTPOS, we generate new queries according to the distinct value combination in the base result set based on the determining set of the constrained attribute. However, these queries may not be equally good in terms of retrieving relevant uncertain answers. Continuing our running example, though the rewritten queries ($Q_1{:}\,\sigma_{model=A4}$, $Q_2{:}\,\sigma_{model=Z4}$ and $Q_3{:}\,\sigma_{model=Boxster}$) are likely to retrieve uncertain answers that are more relevant to Q than a random tuple with missing $body\ style$ value, they may not be equally good. For instance, based on the value dis-



(a) Precision-Recall Curves (Cars)



(b) Avg. Precision for Top K Tuples Over 10 Body Style Queries

tribution in the sample database, we may find that a $Z4$ model car is more likely to be a *convertible* than a car with $A4$ model. Therefore we build AFD-enhanced Naïve Bayes Classifiers (NBC) which give the probability values $P(body\ style{=}convt\ |model{=}A4)$, $P(body\ style{=}convt\ |model{=}Z4)$ and $P(body\ style{=}convt|model{=}Boxster)$. The NBC uses AFDs as a feature selection step before classification as feature selection has shown to improve classification accuracy[1]. Using these probability values, we rank the rewritten queries according to the relevance of their expected query results. The relevant uncertain answers retrieved by these queries need not be ranked again as they are implicitly ranked based on the rank of the rewritten query that retrieved them.

Although we described the above algorithm in the context of single attribute selection queries, our algorithms have been adapted and generalized to support general queries involving selection, joins and aggregations [4].

3 Empirical Evaluation for QPIAD

QPIAD system is implemented in Java and has a form based query interface. The system returns each relevant uncertain answer to the user along with a *confidence* measure equal to the assessed degree of relevance. *QPIAD* can also optionally "explain" its relevance assessment by providing snippets of its reasoning. In particular, it justifies the confidence associated with an answer by listing the AFD that was used in making the relevance assessment.

We compare the effectiveness of RELEVANTPOS with ALLPOS and RANKEDPOS in terms of retrieving relevant uncertain answers. The RANKEDPOS approach first retrieves all the certain and uncertain answers as in ALLPOS, then it ranks uncertain answers according to the probability distribution computed using the NBC classifier.

The effectiveness is measured by precision and recall with respect to uncertain answers at the time when the mediator sees the K^{th} ($K{=}1, 2, 3, \cdots$) answer tuple on randomly formulated selection queries. Figure 2(a) shows the precision and recall curves of a query on a Cars database extracted from Cars.com (www.cars.com). It shows that both RANKEDPOS and RELEVANTPOS approach have significantly higher precision compared to ALLPOS.

Furthermore, the numbers in the parenthesis in the legend in Figure 2(a) are the number of uncertain answers retrieved by each method. As we can see, RELEVANTPOS avoids retrieving too many irrelevant tuples and therefore very efficient. It can further cut down the cost by only sending the top ranked rewritten queries to the database.

To reflect the "density" of the relevant answers along the time line, we also plot the precision of each method at the time when top $K(K{=}1, 2, \cdots, 100)$ answers are retrieved as shown in Figure 2(b). RANKEDPOS and RELEVANTPOS are much better in retrieving relevant uncertain answers in top K results which is critical in web scenarios.

4 Conclusion

Incompleteness is inevitable in autonomous web databases. Retrieving highly relevant uncertain answers from such databases is challenging due to the restricted access privileges of mediator, limited query patterns supported by autonomous databases, and sensitivity of database and network workload in web environment. We developed a novel query rewriting technique that tackles these challenges. Our approach involves rewriting the user query based on the knowledge of database attribute correlations. The rewritten queries are then ranked by leveraging attribute value distributions according to their likelihood of retrieving relevant uncertain answers before they are posed to the databases. Our experiments demonstrated the effectiveness of our query processing and knowledge mining techniques.

Acknowledgements: We thank Garrett Wolf and Bhaumik Chokshi for their helpful comments and enlightening discussions.

References

[1] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 1997.

[2] Y. Huhtala, J. Krkkinen, P. Porkka, and H. Toivonen. Efficient discovery of functional and approximate dependencies using partitions. In *ICDE Conference*, pages 392–401, 1998.

[3] Z. G. Ives, A. Y. Halevy, and D. S. Weld. Adapting to source properties in processing data integration queries. In *SIGMOD Conference*, 2004.

[4] H. Khatri. *Query Processing over Incomplete Autonomous Web Databases,MS Thesis,Dept. of CSE, Arizona State University,* rakaposhi.eas.asu.edu/hemal-thesis.pdf. 2006.

[5] D. Lembo, M. Lenzerini, and R. Rosati. Source inconsistency and incompleteness in data integration. In *KRDB Workshop*, 2002.