ABSTRACT

Databases today are deterministic, that is, an item is either in the database or not. Similarly, a tuple is either in the query result or not. This process of mapping the real world inherently includes ambiguities and uncertainties and is seldom perfect. In today’s data-driven competitive world a wide range of applications have emerged that needs to handle very large, imprecise data sets with inherent uncertainties. Uncertain data is natural in many important real world applications like environmental surveillance, market analysis and quantitative economic research. Data uncertainty innate in these important real world applications is generally the result of factors like data randomness and incompleteness, misaligned schemas, limitations of measuring equipment, delayed data update, imprecise queries etc.[5,11]. Due to the importance of these applications and the rapidly increasing amount of uncertain data collected and accumulated, analyzing large collections of uncertain data has become an important task and has attracted more and more interest from the database community.

Probabilistic Databases hold the promise of being a viable means for large-scale uncertainty management, increasingly being required in a large number of real world application domains [11]. A probabilistic database is an uncertain database in which the possible worlds have associated probabilities, that is, an item belongs to the database is a probabilistic event either with tuple-existence uncertainty or with attribute-value uncertainty. However, a tuple as an answer to query is again a probabilistic event.

An important aspect in tackling the research and development on uncertain data processing is the query answering techniques on uncertain and probabilistic data. Query processing in probabilistic databases remains a computational challenge as it is fundamentally more complex than other data models. There exists a rich collection of powerful, non-trivial techniques and results, some old, some very recent, that could lead to practical management techniques for probabilistic databases. However, all such techniques suffer from limitations of uncertainty inherent in result of the query. Hence, there is a need for a general probabilistic model that tackles this uncertainty at the grass root level.

The basic tool for dealing with this uncertainty is probability which is defined for an event as the proportion of times that the event would occur in repetitions of essentially identical situations. Although useful and successful in many applications, probability theory is, in fact, appropriate for dealing with only a very special type of uncertainty for measuring information. Probabilistic databases are all the more susceptible to uncertainties in query results being exclusively dependent on the probabilities assigned with inherent uncertainty in the evaluation of probabilities. Thus it becomes a potential area where this fundamental problem can be addressed and a suitable correction can be made to probabilities evaluated thereof.

KEYWORDS

Probabilistic Databases, Query optimization, Uncertain Data, Uncertainty, Probability Theory

1. INTRODUCTION

Data in a database is typically treated as being correct and indisputable. In many applications, this obviously is not really true. For example, data may be out of date, or some value may just be the most likely one and could very well be wrong. This is even truer for the results of automatic tasks like information extraction, natural-language processing, data integration, sensor data management, or data mining [12].

To better support such applications, there is growing interest in the management of uncertain data, i.e., data for which we explicitly store the fact that it is uncertain together with information about its uncertainty.

Probabilistic databases are database systems capable of handling large amounts of uncertain data. Most of the common commercial databases today are deterministic rooted in first order logic and finite model theory as envisioned by Codd [2]. These databases are generally used in applications like banking and payroll that require a precise semantics of the data. Databases and data management techniques have been extended to handle richer data models like nested relations Object Relational Data temporal data, spatial data, semi structured and XML. All these extensions rely on a deterministic semantics for both data and queries.

In the present times, the database community needs to manage large amounts of data that is imprecise, uncertain and contains an explicit representation of uncertainty. Uncertain data may occur in large scale integration like integration of life-science databases’ in information extraction systems, in sensor data etc.

Modeling and managing data where uncertainties are explicitly
expressed and numerically quantified requires a new paradigm whose foundation is based on probability theory and logic.

Queries over probabilistic databases can be categorized either as safe, in which case they can be evaluated entirely in a relational database engine or unsafe, in which case they need to be evaluated by a general-purpose inference engine at a higher cost. Evaluating queries over probabilistic databases is in general hard. Despite the recent advances in the present approaches to query evaluation, they are not practical. Existing techniques either split the queries into safe and unsafe and compute efficiently only the former or work very well on certain combinations of queries and data instances but cannot offer performance guarantees in general or use general-purpose approximation techniques and are thus generally slow.

2. PROBABILISTIC DATABASES: PARADIGM FOR MANAGING UNCERTAINTIES

Several approaches have been considered to manage and represent uncertainties e.g. rule-based systems, fuzzy sets but for over the past two decades or more the probability model has been dominant [3]. Probabilistic databases adopt the same model. Data is probabilistic, where the probabilities are internal measures of the imprecision in the data. Probabilistic databases can store types of information that cannot be represented using the relational model. Probabilistic databases may also be viewed as generalizations of relational databases; any relational database can be represented without loss of information by a probabilistic database [2].

Users formulate queries using a standard query language, as if the data were precise. The system computes the answers, and for each answer works out a probability score representing its confidence in that answer. The central problem in probabilistic databases is query evaluation. Given a Boolean query \( q \) and a probabilistic database, compute the answer of \( q \) on the database. On a deterministic database the answer is a Boolean value, true or false, but on a probabilistic database the answer to \( q \) is a probability, in notation \( P(q) \).

The query evaluation problem can be reduced to a special case of inference in a probabilistic network, which is a problem extensively studied in the Knowledge Representation community, and which is known to be hard: both exact inference [11] and approximate inference are NP-hard. The key distinction is that in probabilistic databases we refine the complexity analysis by separating the query from the data, since the size of the data is by far dominant. In light of this distinction, approximate inference is tractable, and, as we explain here, even precise inference is tractable for an important class of queries.

3. PROBABILISTIC DATA MODEL

A probabilistic database is implied in terms of probability distributions on possible worlds, which extends incomplete databases [7]. Throughout this paper we restrict our discussion to relational data over a finite domain.
A Naive Approach For Handling Uncertainty Inherent In Query Optimization of Probabilistic Databases

Techniques from graphical models have also been applied to many topics directly of interest to the database community including information extraction, sensor data analysis, and imprecise data representation and querying, selectivity estimation for query optimization, and data privacy. **Bayesian Networks or Belief Nets** (BNs) are compact models for representing uncertainty in our knowledge of joint distributions over sets of variables \(^{[14]}\). The compactness is achieved by utilizing the conditional independences among the variables. A Bayesian Network consists of a directed acyclic graph (DAG) that includes a node and a conditional probability distribution (CPD) for each variable. The conditional probability distribution of a variable encodes its distribution given its parents in the graph. Thus, for a graph \( G \) with \( N \) nodes \( x = \{ x_1, x_2, \ldots, x_N \} \), the joint distribution is given by

\[
p(x) = \prod_{i=1}^{N} p(x_i | pa(x_i))
\]

where \( pa(x_i) \) denotes the parents of \( x_i \).

The uncertainty in a Bayesian Network may arise in a number of situations like: uncertainty in experts themselves relating to their knowledge; uncertainty inherent in the domain being modeled; uncertainty as to the accuracy and availability of knowledge. Bayesian networks use probability in theory to manage uncertainty by explicitly representing the conditional dependencies between different knowledge components. Thus providing an intuitive graphical visualization of the knowledge including the interactions among the various sources of uncertainty.

Previous work used functional relations to represent the conditional probabilities of a BN in a relational database \(^{[8]}\). A functional relation \( R \) has the schema \( \{ A_1, A_2, \ldots, A_n, f \} \) where \( f \) is called the measure attribute and the functional dependency \( A_1, A_2, \ldots, A_n \rightarrow f \) holds. In this case, the measure attribute corresponds to the conditional probability for the configuration represented by a tuple.

Incorporating probabilities into the semantics of incomplete databases has posed many challenges, forcing systems to sacrifice modeling power, scalability, or treatment of relational algebra operators.

5. INFORMATION THEORY AND UNCERTAINTY
The process of mapping the real world inherently includes ambiguities and uncertainties and is seldom perfect. The basic tool for dealing with this uncertainty is probability which is defined for an event as the probability limiting proportion of times that the event would occur in repetitions of essentially identical situations. Although useful and successful in many applications, probability theory is, in fact, appropriate for dealing with only a very special type of uncertainty for measuring information. A measure of uncertainty when adopted as a measure of information does not include the semantic and pragmatic aspects of information \(^{[1]}\).
Generic Solution for Uncertainty in Probability Theory
Shannon introduced a measure for the purpose of measuring information in terms of uncertainty. This measure he referred to as Shannon entropy. From the analysis of uncertainty measure and the estimation of entropy in information, a generic solution, which is an approximation to the general uncertainty formula, may be written as

\[ I(t) = \sum_{i=1}^{n} Pr(w_i) \ln \frac{1}{Pr(w_i)} \]

(2)

where \( I(t) \) is the information content at an instant of time \( t \). The value of the Information content thus determined, is proportionately back transformed to yield the modified value of the outcome: \( C' \).

6. PROBABILISTIC DATABASES AND UNCERTAINTY
Uncertainty being omnipresent, especially in probabilistic databases modeling uncertainty is a necessary component of these databases. A probabilistic database PDB is a set of possible worlds PDB = \{w_1, . . . ,w_n\} each with its probability \( Pr(w_i) \) such that \( \sum_{i=1}^{n} Pr(w_i) = 1 \). The frequentist view of probability however takes the perspective that probability is an objective concept. The probability of an event is defined as the limiting proportion of times that the event would occur in repetitions of essentially identical situations. Hence, the semantics of a query on a probabilistic database are defined as the set of answers of the query posed to each of the possible worlds individually. Consequently, the probability of a particular answer is the sum of the probabilities of all possible worlds for which the query produced that answer [6].

Since the number of possible worlds grows exponentially, implementations of probabilistic databases store all possible worlds in one compact representation. Algorithms for querying a probabilistic database directly work on the compact representation while strictly adhering to the semantics of querying as defined in terms of possible worlds.

7. MAPPING UNCERTAINTY CONSIDERATION OF INFORMATION THEORY TO PROBABILISTIC DATABASES
Database research has primarily concentrated on how to store and query exact data [3]. These efforts have led to the development of techniques that allow the user to express and efficiently process complex queries in a declarative fashion over large sets of data. However, many real world applications produce large amounts of uncertain data [4]. In such a situation, databases need to do more than simply store and retrieve; they have to help the user sift through the uncertainty and find the tuples most likely to be the answer.

Many approaches have been proposed to answer this inherent uncertainty in databases [12]. However, we can model the uncertainty of probabilistic databases by applying the uncertainty considerations of information theory to probabilistic databases.

8. QUERY EVALUATION CHALLENGES IN PROBABILISTIC DATABASES
A major challenge in use of probabilistic databases is the evaluation of SQL queries on probabilistic databases [8]. Due to the uncertainty, a major fundamental challenge is how to handle the probabilities of uncertain and probabilistic data. In a certain case, a range query on certain data will returns tuples falling in the given range. However, on uncertain and probabilistic data, a tuple may take a probability to fall into a given range. Hence, this probability of being the part of the result is a new dimension that does not appear in traditional queries on certain data.

Another issue with queries on uncertain and probabilistic data is the fact that some queries are only concerned with the uncertainty within individual objects or tuples involved in individual generation rules. Such queries are said to involve local uncertainty. On the other hand, some queries on the same data may involve global uncertainty, which is the uncertainty of combinations of objects/tuples being a part of result set in possible worlds.

For a query on certain data which can be answered efficiently (i.e., in polynomial time), an extension of the query on uncertain and probabilistic data may be in nature of exponential time complexity. Ranking queries are an example.

If an extension has to consider the global uncertainty by examining the possible worlds, due to the exponential number of possible worlds, the problem is #P-complete.

To address the computational challenge from queries on uncertain and probabilistic data, we need to explore the tradeoff between accuracy and computational cost.

Query evaluation is the hardest technical challenge in a probabilistic database management system. One approach is to separate the query and lineage evaluation from the probabilistic inference on the lineage expression. Various algorithms have been used for the latter. Another approach is to integrate probabilistic inference with the query computation step. The advantage of this approach is that it allows us to leverage standard data management techniques to speed up the probabilistic inference, such as static analysis on the query or using materialized views. This has led to safe queries and to partial representations of materialized views.

As an illustration, we consider a simple query based on the probabilistic database in Fig. 1.

```sql
select distinct * from Q1(*)::-- Product(x,y,z) MobileDetail, Order, Order(x,y,u) Customer where Customer(u,v) MobileDetail.prod=Order. prod and MobileDetail.price=Order. price and Order.cust=Customer.cust
```

Fig. 3: Simple Query expressed in SQL and Datalog
The above query is a standard query, i.e. they are written assuming that the database is deterministic, and ignore any probabilistic information. However the semantics of the query is modified as each tuple returned has an associated probability representing our confidence in the answer.

In general a given query Q and a tuple t, the probability that t is an answer to Q is the sum of probabilities of all possible worlds where Q returns t.

9. PROPOSED UNCERTAINTY CORRECTION TO QUERY OPTIMIZATION

Evaluating queries over probabilistic databases (PDBs) is hard in general. Despite important recent advances, today's approaches to query evaluation are not practical. Existing techniques either split the queries into safe or unsafe and compute efficiently only the former, or work very well on certain combinations of queries and data instances but cannot offer performance guarantees in general, or use general purpose approximation techniques and are thus generally slow.

The semantics of a query over a probabilistic database is based on reliability, which has roots in network reliability. It is defined as the probability that a source node remains connected to a target node in a directed graph if edges fail independently with known probabilities. Taking this as the guiding factor we suggest the application of the generic solution to uncertainty in information theory to the uncertainty in probabilistic databases. Due to the generic form of the uncertainty equation, it is extensible to the applications of probabilistic databases that are in the real field: (0, 1).

Considering Certainty \[ C(t) \] as probability of inclusion of tuples in the result set at a certain instant of time \( t \) in a probabilistic database, the complement of certainty is the uncertainty, which can be expressed as:

\[ UC(t) = C(t) \]

(3)

Where \( C(t) \) can be expressed as a sum of all possible probabilities. Keeping in view from the above discussion that information of possible worlds is susceptible to high degree of inherent uncertainties we make a correction suggested by the formula which was an inference deduced from Shannon entropy in information theory in [1] as follows:

Probability of tuple with uncertainty correction \( = C(t)* \) which is given by:

\[ C(t)^* = 1 - UC(t) \]

(4)

Uncertainty evaluation occurs after result data and lineage have been computed during the data processing phase. It involves computing mass, belief, or plausibility values for a result tuple, henceforth collectively referred to as uncertainty values.

This correction can be further extended to the concepts, algorithms and techniques of query optimization. For instance, consider the concept for implementation of complex selections:

The selectivity of a condition \( \theta_i \) for conjunction \( (\sigma \theta_1 \land \theta_2 \land \ldots \land \theta_n(r)) \) is the probability that a tuple in the relation \( r \) satisfies \( \theta_i \). If \( si \) is the number of satisfying tuples in \( r \), the selectivity of \( \theta_i \) is given by \( si/|r| \).

The estimate for number of tuples in the result for a deterministic database is given by:

\[ n_r' = S_1 \times S_2 \times \ldots \times S_n \]

(5)

In case of a probabilistic database the same may be given by

\[ n_r = (1 - (1 - \frac{s_1}{n_r}) \times (1 - \frac{s_2}{n_r}) \times \ldots \times (1 - \frac{s_n}{n_r})) \]

(6)

where \( p_i \) denotes the sum of individual probabilities of each tuple in an \( s_i \).

In a deterministic database the estimated number of tuples for disjunction \( (\sigma \theta_1 \lor \theta_2 \lor \ldots \lor \theta_n(r)) \) is given by:

\[ n_r = \left( 1 - \frac{s_1}{n_r} \right) \times \left( 1 - \frac{s_2}{n_r} \right) \times \ldots \times \left( 1 - \frac{s_n}{n_r} \right) \]

(7)

In case of a probabilistic database the same may be given by

\[ n_r = (1 - (\frac{s_1}{n_r})) \times (1 - (\frac{s_2}{n_r}) \times \ldots \times (1 - (\frac{s_n}{n_r})) \]

(8)

where \( p_i \) denotes the sum of individual probabilities of each tuple in an \( s_i \).

10. CONCLUSION

In many applications today it is becoming prohibitively expensive, and even impossible, to enforce precise semantics, by completely cleaning the data and removing all types of uncertainties. Probabilistic databases manage data with an explicit representation and quantification of uncertainties. However, query evaluation for such databases becomes a special form of probabilistic inference with many different computational complexities.

This paper addresses the above problem by suggesting a generic correction to be applied to any of the traditional query optimization algorithms that are applied to probabilistic data by drawing inferences from information theory. Consequently, an attempt is made to suggest an appropriate correction to the fundamental probabilities assigned to the tuples in the database. We also formulate a simple procedure to evaluate the number of tuples involving conjunctions and disjunctions in the implementation of complex queries which takes into account probabilities suitably corrected for uncertainties.

11. FUTURE SCOPE

This generic correction formula can be extended to further areas of query optimization such as search algorithms and join selections wherein probabilities can be treated as weights to provide expected values. This approach is also applicable to various types of databases such as knowledge, distributed, deductive, object-oriented databases etc. However incorporation of this correction varies subject to database under consideration. The complete scope can be extended to almost every area of computer science which is dependent on probabilities assigned to information. In an age of complex technological advancements and their applicability to
increasing demands of the industry there is an increasing trend towards quantifying intangible aspects to reduce the vagueness and ambiguities inherent in the information and its usage. Hence, it becomes imperative to challenge our fundamental thought processes and think new.

REFERENCES

[4]. P. Sen and A. Deshpande- Representing and querying correlated tuples in probabilistic databases,ICDE’07.
[5]. D. Suciu and N. N. Dalvi,- Foundations of probabilistic answers to queries, SIGMOD’05, page 963.