Query Answering Techniques on Uncertain and Probabilistic Data

Tutorial Summary

Jian Pei[†] Ming Hua[†] Yufei Tao[‡] Xuemin Lin[¶]
[†]Simon Fraser University, Canada
[‡]The Chinese University of Hong Kong, China
[¶]The University of New South Wales, Australia
{jpei, mhua}@cs.sfu.ca, taoyf@cse.cuhk.edu.hk, lxue@cse.unsw.edu.au

ABSTRACT

Uncertain data are inherent in some important applications, such as environmental surveillance, market analysis, and quantitative economics research. Due to the importance of those 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. Recently, uncertain data management has become an emerging hot area in database research and development. In this tutorial, we systematically review some representative studies on answering various queries on uncertain and probabilistic data.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems

General Terms

Algorithm, Performance

Keywords

Uncertain Data, Probabilistic data, Query processing

1. OBJECTIVES AND SCOPE

Uncertain data are inherent in some important applications, such as environmental surveillance, market analysis, and quantitative economics research. Uncertain data in those applications are generally caused by factors like data randomness and incompleteness, limitations of measuring equipment, delayed data updates, etc. Due to the importance of those 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.

In the last several years, some exciting progress has been achieved in the research and development on uncertain data management (e.g., [33, 6, 20, 2, 17, 16, 19, 5, 10, 21]). Many database researchers are joining the workforce to tackle the grand challenges in large scale uncertain data processing. To understand the challenges and the opportunities in the research and development on uncertain and probabilistic data management, we present a tutorial on a fundamental aspect – query answering techniques on uncertain and probabilistic data. We will briefly review the models of uncertain and probabilistic data representation in databases and the possible worlds semantics, and go deep into query answering techniques including algorithms and index structures for various types of queries such as range search queries and ranking queries. We will also discuss some interesting directions for future work.

2. MODELS AND POSSIBLE WORLDS

We consider uncertain data in the *possible worlds* semantics model [1, 24, 33, 16], which has been extensively adopted by the recent studies on uncertain data processing, such as [36, 6, 29]. Technically, uncertain data can be represented in two ways as discussed in Sections 2.1 and 2.2, respectively.

2.1 The Probabilistic Database Model

A *probabilistic database* [33, 36, 6] is a finite set of probabilistic tables. Generally, a *probabilistic table* T contains a set of (uncertain) tuples, where each tuple $t \in T$ is associated with a *membership probability* value $Pr(t) > 0$.

A *generation rule* on a table T specifies a set of exclusive tuples in the form of $R : t_{r_1} \oplus \cdots \oplus t_{r_m}$ where $t_{r_i} \in T$ $(1 \leq i \leq m)$ and $\sum_{i=1}^{m} Pr(t_{r_i}) \leq 1$. The rule R constrains that, among all tuples $t_{r_1}, \ldots, t_{r_m}$ involved in the rule, at most one tuple can appear in a possible world. We denote by $|R| = m$ the number of tuples involved in R. We also say $t_{r_i} \in R$. A generation rule R is a *singleton rule* if there is only one tuple involved in the rule (i.e., $|R| = 1$), otherwise, R is a *multi-tuple rule* (i.e., $|R| > 1$).

The *existence probability* of a generation rule R is the probability that one tuple involved in R appears. That is, $P(R) = \sum_{t \in R} Pr(t)$.

Let $\mathcal{R}_T$ be the set of generation rules on table T in question. For any two rules $R_1, R_2 \in \mathcal{R}_T$, we assume that R_1 and R_2 do not share any common tuple, i.e., $R_1 \cap R_2 = \emptyset$.

For a subset of tuples $S \subseteq T$ and a generation rule R, we denote the tuples involved in R and appearing in S as $R \cap S$. A *possible world* W is a subset of T such that for each generation rule $R \in \mathcal{R}_T$, W contains exactly one tuple involved in R (i.e., $|R \cap W| = 1$) if $Pr(R) = 1$, and W contains 0 or 1 tuple involved in R (i.e., $|R \cap W| \leq 1$) if $Pr(R) < 1$. We denote by $\mathcal{W}$ the set of all possible worlds.

SIGMOD'08, June 9–12, 2008, Vancouver, BC, Canada.
ACM 978-1-60558-102-6/08/06.

RID	Loc.	Time	Sensor-id	Temperature	Conf.
$R1$	A	6/2/06 2:14	$S101$	25	0.3
$R2$	B	7/3/06 4:07	$S206$	21	0.4
$R3$	B	7/3/06 4:09	$S231$	13	0.5
$R4$	A	4/12/06 20:32	$S101$	12	1.0
$R5$	E	3/13/06 22:31	$S063$	17	0.8
$R6$	E	3/13/06 22:28	$S732$	11	0.2

Table 1: Temperature records.

Possible world	Probability	Top-2 on Duration
$W1 = \{R1, R2, R4, R5\}$	0.096	$R1, R2$
$W2 = \{R1, R2, R4, R6\}$	0.024	$R1, R2$
$W3 = \{R1, R3, R4, R5\}$	0.12	$R1, R5$
$W4 = \{R1, R3, R4, R6\}$	0.03	$R1, R3$
$W5 = \{R1, R4, R5\}$	0.024	$R1, R5$
$W6 = \{R1, R4, R6\}$	0.006	$R1, R4$
$W7 = \{R2, R4, R5\}$	0.224	$R2, R5$
$W8 = \{R2, R4, R6\}$	0.056	$R2, R4$
$W9 = \{R3, R4, R5\}$	0.28	$R5, R3$
$W10 = \{R3, R4, R6\}$	0.07	$R3, R4$
$W11 = \{R4, R5\}$	0.056	$R5, R4$
$W12 = \{R4, R6\}$	0.014	$R4, R6$

Table 2: The possible worlds of Table 1.

Clearly, for an uncertain table T and a set of generation rules $\mathcal{R}_T$, the number of all possible worlds is

$$|\mathcal{W}| = \prod_{R\in\mathcal{R}_T, Pr(R)=1} |R| \prod_{R\in\mathcal{R}_T, Pr(R)<1} (|R|+1)$$

The number of possible worlds on a large table can be huge.

Each possible world is associated with an *existence probability* $Pr(W)$ that the possible world happens. Following with the basic probability principles, we have

$$Pr(W) = \prod_{R\in\mathcal{R}_T, |R\cap W|=1} Pr(R\cap W) \prod_{R\in\mathcal{R}_T, R\cap W=\emptyset} (1-Pr(R))$$

Apparently, for a possible world W, $Pr(W) > 0$. Moreover, $\sum_{W\in\mathcal{W}} Pr(W) = 1$.

EXAMPLE 1. *Sensors are often used to monitor environment conditions in remote areas. Due to limitations of sensors, detections cannot be accurate all the time. Instead, detection confidence is often estimated. Table 1 lists a set of synthesized records of temperature detected by sensors.*

In some locations such as B and E, multiple sensors may be deployed to improve the detection quality. Two sensors in the same location (e.g., $S206$ and $S231$, as well as $S063$ and $S732$ in Table 1) may detect the temperature at the (approximately) same time, such as records $R2$ and $R3$, as well as $R5$ and $R6$. In such a case, if the temperature detected by the multiple sensors are inconsistent, at most one sensor can be correct.

The uncertain data in Table 1 carries the possible worlds semantics [1, 24, 16, 33]. The data can be viewed as the summary of a set of possible worlds. The possible worlds are governed by some underlying generation rules which constrain the presence of tuple instances. In Table 1, the fact that $R2$ and $R3$ cannot be true at the same time can be captured by a generation rule $R2 \oplus R3$. Another generation rule is $R5 \oplus R6$.

Table 2 shows all possible worlds and their existence probability values. For example, possible world W_1 contains 4 tuples $R1, R2, R4, R5$. The existence probability of $W1$ is calculated as $Pr(W1) = Pr(R1) \times Pr(R2) \times Pr(R4) \times Pr(R5) = 0.3 \times 0.4 \times 1.0 \times 0.8 = 0.096$.

Since $Pr(R4) = 1.0$, $R4$ appears in every possible world. Moreover, in generation rule $R5 \oplus R6$, $Pr(R5 \oplus R6) = Pr(R5) + Pr(R6) = 0.8 + 0.2 = 1.0$. Thus, in every possible world, either $R5$ or $R6$ appears, but not both. On the other hand, for generation rule $R2 \oplus R3$, $Pr(R2 \oplus R3) = Pr(R2) + Pr(R3) = 0.9 < 1$, in some possible worlds such as $W5, W6, W11, W12$, neither $R2$ nor $R3$ appears.

It can be verified that the sum of existence probabilities of all possible worlds is exactly 1.0. ■

2.2 The Uncertain Object Model

An *uncertain object* [11, 13, 37, 29] is conceptually described by a probability density function (PDF) f in the data space D. Generally, $f(u) \geq 0$ for any point u in the data space D, and $\int_{u\in D} f(u)du = 1$.

Practically, the probability density function of an uncertain object is often unavailable explicitly. Instead, a set of samples are drawn or collected in the hope of approximating the probability density function. Correspondingly, we model an uncertain object U as a set of multiple points in the data space as its instances, denoted by $U = \{u_1, \ldots, u_l\}$. It can be regarded as the *discrete case*. Let the probability mass function (pmf) of U be f, then $f(u_i) > 0$ $(1 \leq i \leq l)$, and $\sum_{1\leq i\leq l} f(u_i) = 1$. The number of instances of an uncertain object U is denoted by $|U| = l$.

Let $X_1, \ldots, X_n$ be n uncertain objects. A possible world $W = \{x_1, \ldots, x_n\}$ contains one instance of each object, i.e., x_i is an instance of X_i $(1 \leq i \leq n)$. The existence probability of a possible world $Pr(W) = \prod_{i=1}^{n} f_i(x_i)$, where f_i is the probability density or mass function of object X_i.

Clearly, $Pr(W) > 0$ for any possible world. Moreover, $\sum_{W\in\mathcal{W}} Pr(W) = 1$.

2.3 Converting between the Two Models

The uncertain object model and the probabilistic database model are equivalent in the discrete case, since a set of uncertain objects can be represented by a probabilistic table as follows. For each instance u of an uncertain object U, we create a tuple t_u, whose membership probability is $f(u)$. For each uncertain object $U = \{u_1, \ldots, u_l\}$, we create one generation rule $R_U = t_{u_1} \oplus \cdots \oplus t_{u_l}$.

In all cases, a probabilistic table can be represented by a set of uncertain objects with discrete instances. For each tuple t in a probabilistic table T, we create an instance u_t, whose probability mass function is $f(u_t) = Pr(t)$. For a generation rule $R : t_{r_1} \oplus \cdots \oplus t_{r_m}$, we create an uncertain object U_R, which includes all instances $u_{t_{r_1}}, \ldots, u_{t_{r_m}}$ corresponding to $t_{r_1}, \ldots, t_{r_m}$, respectively. Moreover, if $\sum_{i=1}^{m} Pr(t_{r_i}) < 1$, we create another instance $u_\emptyset$ whose probability mass function is $f(u_\emptyset) = 1 - \sum_{i=1}^{m} Pr(t_{r_i})$, and add $u_\emptyset$ to the uncertain object U_R. Since any two generation rules do not share any common tuples, the uncertain objects constructed as such do not share any common instances.

3. CHALLENGES OF QUERY ANSWERING ON UNCERTAIN AND PROBABILISTIC DATA

Answering various queries on certain data has been studied extensively and systematically. Then, what are the new challenges that queries on uncertain and probabilistic data pose?

3.1 Probabilities: A New Dimension

Due to the uncertainty, one fundamental challenge is how to handle the probabilities of uncertain data and probabilistic data. For example, on certain data, a range query returns the points falling into a given range. On uncertain and probabilistic data, a tuple or an uncertain object may take a probability to fall into a given range. Consequently, a range query may be extended to uncertain and probabilistic data in multiple ways. First, we can output only those tuples/objects which are absolutely (i.e., with probability 100%) falling into the query range. Second, we can output those tuples/objects which have a non-zero probability to fall into the query range. More generally, we may use a threshold to control the probability requirement on the tuples/objects in the answer set meeting the query range – only those tuples/objects whose probability falling into the query range passing the threshold are returned.

The probability of being an answer is a new dimension that does not appear in conventional query answering on certain data. A few studies use probability thresholds to extend conventional queries to probabilistic threshold queries, such as range search queries [11, 13, 37, 38], ranking queries [22, 23, 27, 40], and skyline queries [30]. In some other methods, results are ranked according to their probabilities of being answers, such as U-Top$k$ queries and U-kRanks queries [36] and the ranking queries in [31].

3.2 Global and Local Uncertainty

In some queries on uncertain and probabilistic data, we are only concerned with the uncertainty within individual objects or tuples involved in individual generation rules. We call such queries involving *local uncertainty*.

For example, in a range search query on uncertain data, the probability whether a tuple/object falls into the query range and thus in the answer set depends on the uncertainty of the tuple/object itself, and is independent from other objects/tuples.

On the other hand, we may have to consider *global uncertainty* when answering some queries on uncertain and probabilistic data, which is the uncertainty of combinations of objects/tuples being answers in possible worlds. For example, the ranking of an uncertain object or a probabilistic tuple in an uncertain data set depends on not only the values of the instances of the object/tuple, but also the values of the instances of the other objects/tuples.

Generally, when whether an object/tuple satisfies a query depends on other objects or tuples not involved in the same generation rule, global uncertainty has to be considered. Semantically, we have to examine the possible worlds one by one and count the probability that a combination of objects/tuples is an answer.

Therefore, assigning proper semantics to queries on uncertain and probabilistic data is a new challenge.

3.3 Computational Cost: Enumeration or Not

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 [15].

To address the computational challenge from queries on uncertain and probabilistic data, we need to explore the tradeoff between accuracy and computational cost. Sampling-based methods and randomized algorithms are particularly interesting since they may provide good quality guarantees on approximate answers for expensive queries, at the same time, remain polynomial in computational cost.

In the next two sections, we will use range search queries and ranking queries on uncertain and probabilistic data to elaborate the challenges discussed above and some representative solutions.

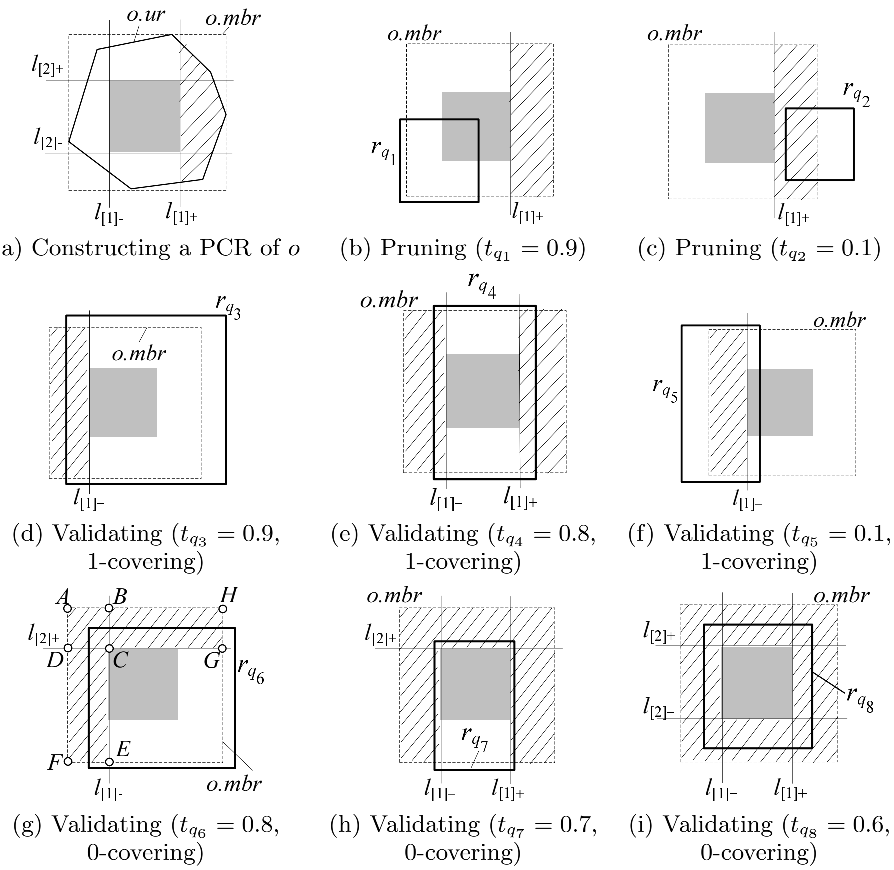
4. RANGE SEARCH QUERIES

Cheng *et al.* [11] provided a general classification of probabilistic queries and evaluation algorithms over uncertain data sets. Different from query answering in traditional data sets, a probabilistic quality estimate was proposed to evaluate the quality of results in probabilistic query answering.

Particularly, *probability-thresholding range queries* [11, 13, 37, 38] are an essential type of queries. Formally, given a region r_q and a probability threshold t_q, such a query returns all objects that appear in r_q with at least probability t_q.

Cheng *et al.* [13] proposed the notion of *x-bounds*, which leads to an efficient access method, called the *probability threshold index* (PTI), for managing one-dimension uncertain data. Tao *et al.* develop the U-tree [37, 38] which extends PTI to index multidimensional uncertain objects. Here, each object o is represented by an *uncertainty region* $o.ur$ and a pdf $o.pdf(x)$. Specifically, $o.ur$ defines the area of the data space where o can possibly appear, while, for any location x in the data space, $o.pdf(x)$ equals the probability that o appears at x (the function $o.pdf(.)$ can also be a probability mass function as described in Section 2.2). As a special case, $o.pdf(x) = 0$ if x is outside $o.ur$. Next, we discuss the U-tree in the context of probability-thresholding range queries.

The U-tree is based on the concept *probabilistically constrained region* (PCR) (which generalizes the conventional x-bounds [13]). A PCR of an object o depends on a parameter $c \in [0, 0.5]$, and hence, is represented as $o.pcr(c)$. It is a d-dimensional rectangle, obtained by pushing, respectively, each face of $o.mbr$ inward, until the appearance probability of o in the area swept by the face equals c. Figure 1a illustrates the construction of a 2D $o.pcr(c)$, where the polygon represents the uncertainty region $o.ur$ of o, and the dashed rectangle is the MBR of o, denoted as $o.mbr$. The $o.pcr(c)$, which is the grey area, is decided by 4 lines $l_{[1]+}$, $l_{[1]-}$, $l_{[2]+}$, and $l_{[2]-}$. Line $l_{[1]+}$ has the property that, the appearance probability of o on the right of $l_{[1]+}$ (i.e., the hatched area) is c. Similarly, $l_{[1]-}$ is obtained in such a way that the appearance likelihood of o on the left of $l_{[1]-}$ equals c (it follows that the probability that o lies between $l_{[1]-}$ and $l_{[1]+}$ is $1 - 2c$). Lines $l_{[2]+}$ and $l_{[2]-}$ are obtained in the same way, except that they horizontally partition $o.ur$.



(a) Constructing a PCR of o (b) Pruning ($t_{q_1} = 0.9$) (c) Pruning ($t_{q_2} = 0.1$)

(d) Validating ($t_{q_3} = 0.9$, 1-covering) (e) Validating ($t_{q_4} = 0.8$, 1-covering) (f) Validating ($t_{q_5} = 0.1$, 1-covering)

(g) Validating ($t_{q_6} = 0.8$, 0-covering) (h) Validating ($t_{q_7} = 0.7$, 0-covering) (i) Validating ($t_{q_8} = 0.6$, 0-covering)

Figure 1: Pruning/validating with a 2D probabilistically constrained rectangle

PCRs can be used to prune or validate an object, without computing its accurate qualification probability. Let us assume that the grey box in Figure 1a is the *o.pcr*(0.1) of o. Figure 1b shows the same PCR and *o.mbr* again, together with the search region r_{q_1} of a range query q_1 whose probability threshold t_{q_1} equals 0.9. As r_{q_1} does not fully contain *o.pcr*(0.1), we can immediately assert that o cannot qualify q_1. Indeed, since o falls in the hatched region with probability 0.1, the appearance probability of o in r_{q_1} must be smaller than $1 - 0.1 = 0.9$. Figure 1c illustrates pruning the same object with respect to another query q_2 having $t_{q_2} = 0.1$. This time, o is disqualified because r_{q_2} does not intersect *o.pcr*(0.1) (the pruning conditions are different for q_1 and q_2). In fact, since r_{q_2} lies entirely on the right of $l_{[1]+}$, the appearance probability of o in r_{q_2} is definitely smaller than 0.1.

The second row of Figure 1 presents three situations where o can be validated using *o.pcr*(0.1), with respect to queries q_3, q_4, q_5 having probability thresholds $t_{q_3} = 0.9$, $t_{q_4} = 0.8$, and $t_{q_5} = 0.1$, respectively. In Figure 1d (or Figure 1f), o must satisfy q_3 (or q_5) due to the fact that r_{q_3} (or r_{q_5}) fully covers the part of *o.mbr* on the right (or left) of $l_{[1]-}$, which implies that the appearance probability of o in the query region must be at least $1 - 0.1 = 0.9$ (or 0.1), where 0.1 is the likelihood for o to fall in the hatched area. Similarly, in Figure 1e, o definitely qualifies q_4, since r_{q_4} contains the portion of *o.mbr* between $l_{[1]-}$ and $l_{[1]+}$, where the appearance probability of o equals $1 - 0.1 - 0.1 = 0.8$.

The queries in Figures 1d-1f share a common property: the projection of the search region contains that of *o.mbr* along one (specifically, the vertical) dimension. Accordingly, we say that those queries *1-cover o.mbr*. In fact, validation is also possible, even if a query *0-covers o.mbr*, namely, the projection of the query area does not contain that of *o.mbr* on any dimension. Next, we illustrate this using the third row of Figure 1, where the queries q_6, q_7, q_8 have probability thresholds $t_{q_6} = 0.8$, $t_{q_7} = 0.7$, and $t_{q_8} = 0.6$, respectively.

In Figure 1g, o is guaranteed to qualify q_6, since r_{q_6} covers entirely the part of *o.mbr* outside the hatched area. Observe that the appearance probability of o in the hatched area is *at most* 0.2. To explain this, we decompose the area into three rectangles $ABCD$, $DCEF$, $BCGH$, and denote the probabilities for o to lie in them as ρ_{ABCD}, ρ_{DCEF}, and ρ_{BCGH}, respectively. By the definition of $l_{[1]-}$, we know that $\rho_{ABCD} + \rho_{DCEF} = 0.1$, whereas, by $l_{[2]+}$, we have $\rho_{ABCD} + \rho_{BCGH} = 0.1$. Since ρ_{ABCD}, ρ_{DCEF}, and ρ_{BCGH} are nonnegative, it holds that $\rho_{ABCD} + \rho_{DCEF} + \rho_{BCGH} \leq 0.2$. This, in turn, indicates that o falls in r_{q_6} with probability at least 0.8 ($= t_{q_6}$). With similar reasoning, it is not hard to verify that, in Figure 1h (Figure 1i), the appearance probability of o in the hatched area is at most 0.3 (0.4), meaning that o definitely satisfies q_7 (q_8).

For each object, the U-tree pre-computes several PCRs at a selected set of probability values, which constitute the *U-catalog*. These PCRs are then organized in a fashion similar to an R-tree.

5. RANKING QUERIES

Arguably, ranking queries are among the most popularly

RID	$R1$	$R2$	$R3$	$R4$	$R5$	$R6$
Probability	0.3	0.4	0.38	0.202	0.704	0.014

Table 3: The top-2 probability values of records in Table 1.

Query type ($k = 2$)	Answer
U-Top$K$ query	$\langle R5, R3\rangle$
U-kRanks query	$\langle R5, R5\rangle$
PT-k query ($p = 0.35$)	$\{R2, R3, R5\}$

Table 4: The answers to three types of top-k queries on Table 1.

used queries in databases. Several proposals of extending ranking queries to uncertain data have been developed and the corresponding algorithms have been devised.

5.1 Query Types

In [36], Soliman *et al.* proposed U-Top$k$ queries and U-kRanks queries. A U-Top$k$ query returns a k-tuple sorted list which has the highest probability to be the top-k list in possible worlds. A U-kRanks query finds the tuple of the highest probability at each ranking position. Thus, the tuples returned by a U-kRanks query may not be a valid top-k tuple list in any possible world, and a tuple may appear more than once in the answer set.

In Table 1, for the U-Top$k$ query, $\langle R5, R3\rangle$ should be returned if $k = 2$. For the U-kRanks query, $\langle R5, R5\rangle$ should be returned if $k = 2$, since $R5$ has the highest probability to be ranked first in all possible worlds, and also has the highest probability to be ranked second in all possible worlds.

Recently, Hua *et al.* tackled probabilistic threshold top-k queries (PT-k query for short) on probabilistic data [22, 23]. Given a probability threshold p ($0 < p \leq 1$) and a parameter k, a PT-k query finds the set of tuples whose probabilities to be in the top-k list are at least p.

In the case of Table 1, the probability that a tuple is in the top-2 lists of all possible worlds is shown in Table 3. If $k = 2$ and $p = 0.35$, then $\{R2, R3, R5\}$ should be returned. Table 4 shows the answers to those queries in this example.

The above extensions use an objective function to rank probabilistic tuples. The critical differences among them are on how ranking results should be captured.

Another category of extensions is to rank tuples based on their probabilities of being answers. Particularly, in [31], Ré *et al.* considered arbitrary SQL queries and the ranking is on the probability that a tuple satisfies the query. Moreover, Zhang and Chomicki developed the global top-k semantics on uncertain data which returns k tuples having the largest probabilities in the top-k list, and gave a dynamic programming algorithm [40]. To this extent, [40] ranks tuples using both an objective function and the top-k probabilities. Similarly, Silberstein *et al.* [35] modeled each sensor in a sensor network as an uncertain object whose values follow some unknown distribution. Then, a top-k query in the sensor network returns the top-k sensors such that the probability of each sensor whose values are ranked top-k in any timestep is the greatest. A sampling-based method collects all values in the network as a sample at randomly chosen timesteps, and the answer to a top-k query is estimated using the samples.

5.2 Query Answering Methods

Accompanying with the extensions of ranking queries on uncertain and probabilistic data, several algorithms are devised to answer various ranking queries [36, 22, 23, 40]. Moreover, Yi *et al.* [39] proposed efficient algorithms to answer U-Top$k$ queries and U-kRanks queries. Their algorithm for U-kRanks uses the Poisson binomial recurrence [26]. Lian and Chen developed the spatial and probabilistic pruning techniques for U-kRanks queries [27].

One of the fundamental ideas in those methods is to enumerate and prune possible answers systematically. For promising candidates, those methods (implicitly) search the possible worlds by estimating the probabilities of those promising candidates through considering the relationship between the candidates and other tuples.

For example, consider a PT-k query [22, 23]. For a tuple t and a possible world W such that $t \in W$, whether t is in the top-k list of W, denoted by $t \in Q^k(W)$, depends only on how many other tuples in T ranked higher than t appear in W. Technically, for a tuple $t \in T$, the *dominant set* of t is the subset of tuples in T that are ranked higher than t, i.e., $S_t = \{t' | t' \in T \wedge t' \prec_f t\}$. Moreover, dominant sets have a nice property: for a tuple $t \in T$, the top-k probability of t in T equals the top-k probability of t in S_t.

Using the dominant set property, the PT-k query answering algorithm in [22, 23] scans the tuples in T in the ranking order, and derives the top-k probability of a tuple t based on the tuples preceding t in the ranking order.

Technically, let $L = t_1 \cdots t_n$ be the list of all tuples in table T in the ranking order. Then, in a possible world W, a tuple $t_i \in W$ ($1 \leq i \leq n$) is ranked at the j-th ($j > 0$) position if and only if exactly $(j - 1)$ tuples in the dominant set $S_{t_i} = \{t_1, \ldots, t_{i-1}\}$ also appear in W.

The *position probability* $Pr(t_i, j)$ is the probability that tuple t_i is ranked at the j-th position in possible worlds. Moreover, the *subset probability* $Pr(S_{t_i}, j)$ is the probability that j tuples in S_{t_i} appear in possible worlds.

Trivially, we have $Pr(\emptyset, 0) = 1$ and $Pr(\emptyset, j) = 0$ for $0 < j \leq n$. Then,

$$Pr(t_i, j) = Pr(t_i)Pr(S_{t_{i-1}}, j - 1)$$

Apparently, the top-k probability of t_i is given by

$$Pr^k(t_i) = \sum_{j=1}^{k} Pr(t_i, j) = Pr(t_i) \sum_{j=1}^{k} Pr(S_{t_{i-1}}, j - 1)$$

Particularly, when $i \leq k$, we have $Pr^k(t_i) = Pr(t_i)$.

Using the Poisson binomial recurrence [26], we can show the following. When all tuples are independent (i.e., no multi-tuple generation rules), for $1 \leq i, j \leq |T|$, $Pr(S_{t_i}, 0) = Pr(S_{t_{i-1}}, 0)(1 - Pr(t_i)) = \prod_{j=1}^{i}(1 - Pr(t_i))$, and $Pr(S_{t_i}, j) = Pr(S_{t_{i-1}}, j - 1)Pr(t_i) + Pr(S_{t_{i-1}}, j)(1 - Pr(t_i))$. Using the result, we can compute the top-k probability values efficiently.

In the general case where multi-tuple generation rules present, the generation rules involving multiple tuples are handled by the rule-tuple compression technique. The probability threshold is used to prune tuples whose top-k probability values fail the threshold.

In addition to the exact algorithms, sampling-based approximation methods are explored (e.g., [23, 35]). The central idea is to use a sample to estimate the probabilities. The guarantees of the approximation quality often come from statistics tools such as a sufficiently large sample size determined by applying Chernoff-Hoeffding bound [4] and distribution properties (e.g., Chernoff Bound of Poisson Trials [28]), and two-stage stochastic optimization with recourse [34, 35].

6. FUTURE DIRECTIONS

In addition to range search queries and ranking queries discussed above, several other types of queries are extended to uncertain and probabilistic data, such as joins [3, 12, 25], views [18, 32], spatial queries [14], skyline queries [30], and OLAP queries [7, 8, 9].

As uncertain and probabilistic data have found more and more applications, many interesting problems about management and analysis of uncertain and probabilistic data emerge. Here, we list three interesting directions as examples for future work.

First, it is interesting to extend the well accepted queries on certain data to uncertain and probabilistic data. For example, it is interesting to explore the semantics and efficient algorithms for advanced queries on uncertain and probabilistic data such as k-nearest neighbor search, reverse nearest neighbor search, continuous nearest neighbor search, etc.

Second, it is important to explore novel types of queries unique for uncertain and probabilistic data. For example, U-kRank queries are an example. An interesting idea is to explore how queries can be formed to explore the probability information.

Last, developing efficient algorithms for queries on uncertain and probabilistic data is a critical task. Particularly, many heuristic methods exist for queries on certain data. How can those methods be extended to uncertain and probabilistic data? What kinds of methods are effective on uncertain and probabilistic data and what kinds are not? In addition to experimental methods, theoretical analysis like [15, 17] is highly desirable.

7. REFERENCES

[1] S. Abiteboul, P. Kanellakis, and G. Grahne. On the representation and querying of sets of possible worlds. In *Proceedings of the 1987 ACM SIGMOD international conference on Management of data (SIGMOD'87)*, pages 34–48, New York, NY, USA, 1987. ACM Press.

[2] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. U. Nabar, T. Sugihara, and J. Widom. Trio: A system for data, uncertainty, and lineage. In *VLDB*, pages 1151–1154, 2006.

[3] P. Agrawal and J. Widom. Confidence-aware joins in large uncertain databases. Technical report, Stanford University CA, USA.

[4] D. Angluin and L. G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. In *Proceedings of the ninth annual ACM symposium on Theory of computing (STOC'77)*, pages 30–41, New York, NY, USA, 1977. ACM Press.

[5] D. Barbará, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng.*, 4(5):487–502, 1992.

[6] O. Benjelloun, A. D. Sarma, A. Halevy, and J. Widom. Uldbs: databases with uncertainty and lineage. In *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*, pages 953–964. VLDB Endowment, 2006.

[7] D. Burdick, P. M. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. OLAP over uncertain and imprecise data. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 970–981. VLDB Endowment, 2005.

[8] D. Burdick, P. M. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. Efficient allocation algorithms for olap over imprecise data. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 391–402. VLDB Endowment, 2006.

[9] D. Burdick, A. Doan, R. Ramakrishnan, and S. Vaithyanathan. Olap over imprecise data with domain constraints. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 39–50. VLDB Endowment, 2007.

[10] R. Cavallo and M. Pittarelli. The theory of probabilistic databases. In *VLDB*, pages 71–81, 1987.

[11] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data (SIGMOD'03)*, pages 551–562, New York, NY, USA, 2003. ACM Press.

[12] R. Cheng, S. Singh, S. Prabhakar, R. Shah, J. S. Vitter, and Y. Xia. Efficient join processing over uncertain data. In *CIKM*, pages 738–747, 2006.

[13] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *VLDB*, pages 876–887, 2004.

[14] X. Dai, M. L. Yiu, N. Mamoulis, Y. Tao, and M. Vaitis. Probabilistic spatial queries on existentially uncertain data. In *Advances in Spatial and Temporal Databases, Proceedings of the 9th International Symposium (SSTD'05)*, volume 3633 of *Lecture Notes in Computer Science*, pages 400–417, Angra dos Reis, Brazil, August 2005. Springer.

[15] N. Dalvi and D. Suciu. The dichotomy of conjunctive queries on probabilistic structures. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'07)*, pages 293–302, New York, NY, USA, 2007. ACM Press.

[16] N. Dalvi and D. Suciu. Management of probabilistic data: foundations and challenges. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'07)*, pages 1–12, New York, NY, USA, 2007. ACM.

[17] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, pages 864–875, Toronto, Canada, 2004.

[18] N. N. Dalvi and D. Suciu. Answering queries from statistics and probabilistic views. In *VLDB*, pages 805–816, 2005.

[19] D. Dey and S. Sarkar. A probabilistic relational model and algebra. *ACM Trans. Database Syst.*, 21(3):339–369, 1996.

[20] X. Dong, A. Y. Halevy, and C. Yu. Data integration with uncertainty. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 687–698. VLDB Endowment, 2007.

[21] T. J. Green and V. Tannen. Models for incomplete and probabilistic information. *IEEE Data Eng. Bull.*, 29(1):17–24, 2006.

[22] M. Hua, J. Pei, W. Zhang, and X. Lin. Efficiently answering probabilistic threshold top-k queries on uncertain data (extended abstract). In *Proc. International Conference on Data Engineering (ICDE'08)*, Cancun, Mexico, April 2008.

[23] M. Hua, J. Pei, W. Zhang, and X. Lin. Ranking queries on uncertain data: A probabilistic threshold approach. In *Proc. ACM International Conference on Management of Data (SIGMOD'08)*, Vancouver, Canada, June 2008.

[24] T. Imielinski and J. Witold Lipski. Incomplete information in relational databases. *Journal of ACM*, 31(4):761–791, 1984.

[25] B. Kimelfeld and Y. Sagiv. Maximally joining probabilistic data. In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 303–312, New York, NY, USA, 2007. ACM.

[26] K. Lange. *Numerical analysis for statisticians*. Statistics and computing. 1999.

[27] X. Lian and L. Chen. Probabilistic ranked queries in uncertain databases. In *Proc. 2008 International Conference on Extended Data Base Technology (EDBT'08)*, March 2008.

[28] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, United Kingdom, 1995.

[29] J. Pei, B. Jiang, X. Lin, and Y. Yuan. Probabilistic skylines on uncertain data. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB'07)*, Viena, Austria, September 2007.

[30] J. Pei, B. Jiang, X. Lin, and Y. Yuan. Probabilistic skylines on uncertain data. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 15–26. VLDB Endowment, 2007.

[31] C. Ré, N. Dalvi, and D. Suciu. Efficient top-k query evaluation on probabilistic data. In *Proceedings of the 23nd International Conference on Data Engineering (ICDE'07)*, Istanbul, Turkey, April 2007. IEEE.

[32] C. Re and D. Suciu. Materialized views in probabilistic databases for information exchange and query optimization. In *VLDB*, pages 51–62, 2007.

[33] A. D. Sarma, O. Benjelloun, A. Halevy, and J. Widom. Working models for uncertain data. In *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, page 7, Washington, DC, USA, 2006. IEEE Computer Society.

[34] D. B. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 228–237, Washington, DC, USA, 2004. IEEE Computer Society.

[35] A. S. Silberstein, R. Braynard, C. Ellis, K. Munagala, and J. Yang. A sampling-based approach to optimizing top-k queries in sensor networks. In *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering*, page 68, Washington, DC, USA, 2006. IEEE Computer Society.

[36] M. A. Soliman, I. F. Ilyas, and K. C.-C. Chang. Top-k query processing in uncertain databases. In *Proceedings of the 23nd International Conference on Data Engineering (ICDE'07)*, Istanbul, Turkey, April 2007. IEEE.

[37] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao, and S. Prabhakar. Indexing multi-dimensional uncertain data with arbitrary probability density functions. In *Proceedings of the 31st international conference on Very large data bases (VLDB'05)*, pages 922–933. VLDB Endowment, 2005.

[38] Y. Tao, X. Xiao, and R. Cheng. Range search on multidimensional uncertain data. *ACM Trans. Database Syst.*, 32(3):15, 2007.

[39] K. Yi, F. Li, D. Srivastava, and G. Kollios. Efficient processing of top-k queries in uncertain databases. In *Proc. 2008 International Conference on Data Engineering (ICDE'08)*, April 2008.

[40] X. Zhang and J. Chomicki. On the semantics and evaluation of top-k queries in probabilistic databases. In *Proc. the Second International Workshop on Ranking in Databases (DBRank'08)*, April 2008.