Efficient Processing of Top-k Queries in Uncertain Databases

Ke Yi [#], Feifei Li [†], George Kollios [‡], Divesh Srivastava [§]

[#]*Department of Computer Science and Engineering, Hong Kong University of Science and Technology*
[†]*Department of Computer Science, Florida State University*
[‡]*Department of Computer Science, Boston University,* [§]*AT&T Labs-Research*
[#]yike@cse.ust.hk, [†]lifeifei@cs.fsu.edu, [‡]gkollios@cs.bu.edu, [§]divesh@research.att.com

***Abstract*— This work introduces novel polynomial-time algorithms for processing top-k queries in uncertain databases, under the generally adopted model of x-relations. An x-relation consists of a number of x-tuples, and each x-tuple randomly instantiates into one tuple from one or more alternatives. Our results significantly improve the best known algorithms for top-k query processing in uncertain databases, in terms of both running time and memory usage. Focusing on the single-alternative case, the new algorithms are orders of magnitude faster.**

I. Introduction

Uncertain databases have received a lot of attention recently due to the large number of applications that require management of uncertain and/or fuzzy data. Examples of such applications include: data integration, data cleaning, and mobile and sensor data management.

a) The uncertain data model.: In the TRIO [1] system, an uncertain data set, which they call an *x-relation*, consists of a number of *x-tuples*. Each x-tuple includes a number of alternatives, associated with probabilities, which represent a discrete probability distribution of these alternatives being selected. Independence is still assumed among the x-tuples. In this paper, we also adopt the x-relation model, augmented with a score attribute, on which we rank the tuples. More precisely, each *tuple* t consists of four components: a unique identifier $id(t)$, a *score* $s(t)$, a *confidence* $p(t)$ that is the probability of t appearing in the database, and all the other attributes $A(t)$. An *x-tuple* τ is a finite set of tuples, subject to the constraint that $\sum_{t_i \in \tau} p(t_i) \leq 1$. These t_i's are called the *alternatives* of τ. An x-tuple represents a discrete probability distribution of the possible values τ may make in a randomly instantiated database, i.e., τ takes t_i with probability $p(t_i)$, for $i = 1, \ldots, |\tau|$[1], or does not appear at all with probability $1 - \sum_{i=1}^{d} p(t_i)$. We define an *uncertain database* $\mathcal{D}$ as a collection of M pairwise disjoint x-tuples. We use D to denote the set of all tuples in $\mathcal{D}$, and let $|D| = \sum_{\tau \in \mathcal{D}} |\tau| = N$. Without loss of generality, we assume that all scores are distinct in D.

An uncertain database $\mathcal{D}$ is instantiated into a *possible world* assuming mutual independence of the x-tuples [1] and we let $\mathcal{W}$ be the set of all possible worlds. Thus, $\mathcal{D}$ represents a probability distribution over $\mathcal{W}$ in a succinct format. Please refer to Figure 1 for an example. We distinguish between two cases. In the single-alternative case (e.g., x-tuple τ_2), each x-tuple has only one alternative; in the multi-alternative case (e.g., x-tuple τ_1), there could be more than one alternative for an x-tuple.

[1]We denote the number of alternatives for an x-tuple τ as $d = |\tau|$.

tuples	$s(t)$	$p(t)$
t_1	100	0.5
t_2	92	0.4
t_3	80	0.6
t_4	70	0.3

x-tuples	
τ_1	$\{t_1, t_4\}$
τ_2	$\{t_2\}$
τ_3	$\{t_3\}$

world W	$\Pr[W]$
$\emptyset$	$(1-p(t_1)-p(t_4))(1-p(t_2))(1-p(t_3)) = .048$
$\{t_1\}$	$p(t_1)(1-p(t_2))(1-p(t_3)) = .12$
$\{t_2\}$	$p(t_2)(1-p(t_1)-p(t_4))(1-p(t_3)) = .032$
$\{t_3\}$	$p(t_3)(1-p(t_1)-p(t_4))(1-p(t_2)) = .072$
$\{t_4\}$	$p(t_4)(1-p(t_2))(1-p(t_3)) = .072$
$\{t_1, t_2\}$	$p(t_1)p(t_2)(1-p(t_3)) = .08$
$\{t_2, t_4\}$	$p(t_2)p(t_4)(1-p(t_3)) = .048$
$\{t_1, t_3\}$	$p(t_1)p(t_3)(1-p(t_2)) = .18$
$\{t_3, t_4\}$	$p(t_3)p(t_4)(1-p(t_2)) = .108$
$\{t_2, t_3\}$	$p(t_2)p(t_3)(1-p(t_1)-p(t_4)) = .048$
$\{t_1, t_2, t_3\}$	$p(t_1)p(t_2)p(t_3) = .12$
$\{t_2, t_3, t_4\}$	$p(t_2)p(t_3)p(t_4) = .072$

Fig. 1. An example uncertain database and all its possible worlds.

b) Top-k queries in an uncertain database.: This paper investigates query processing issues under the setting of uncertain data, and in particular we concentrate on top-k queries as defined in [2].

*Definition 1 (*Uncertain Top-k Query (U-Top$k$)): Let $\mathcal{D}$ be an uncertain database with possible worlds space $\mathcal{W}$. For any $W \in \mathcal{W}$, let $\Psi(W)$ be the top-k tuples in W by the score attribute; if $|W| < k$, define $\Psi(W) = \emptyset$. Let T be any set of k tuples. The answer T^* to a U-Top$k$ query on $\mathcal{D}$ is $T^* = \arg\max_T \sum_{W \in \mathcal{W}, \Psi(W)=T} \Pr[W]$. Ties can be broken arbitrarily.

For the example in Figure 1, the U-Top2 answer is $\{t_1, t_2\}$, with a probability of $0.08 + 0.12 = 0.2$, contributed by the world $\{t_1, t_2\}$ and $\{t_1, t_2, t_3\}$ respectively.

*Definition 2 (*Uncertain k-Ranks Query (U-kRanks)): Let $\mathcal{D}$ be an uncertain database with possible worlds space $\mathcal{W}$. For any $W \in \mathcal{W}$, let $\psi_i(W)$ be the tuple with the i-th largest score, for $1 \leq i \leq |W|$. The answer to a U-kRanks query on $\mathcal{D}$ is a vector $(t_1^*, \ldots, t_k^*)$, where $t_i^* = \arg\max_t \sum_{W \in \mathcal{W}, \psi_i(W)=t} \Pr[W]$, for $i = 1, \ldots, k$. Ties can be broken arbitrarily.

	U-Top$k$		U-$k$ranks	
	time	**space**	**time**	**space**
ours	$n \log k$	k	nk	k
[2]	nk	k^2	n^2k	nk

Fig. 2. Asymptotic results for single alternative case in x-relation model, where n is the scan depth.

For the example in Figure 1, the U-2Ranks answer is (t_1, t_3): t_1 has a probability of $0.12+0.08+0.18+0.12 = 0.5$ of being at rank 1, and t_3 has a probability of $0.18 + 0.048 + 0.072 = 0.3$ of being at rank 2.

Focusing on the single-alternative case, we provide solutions for both U-Top$k$ queries and U-kRanks queries, both of which are significantly faster and use much less space under the x-relation model. A comparison of the asymptotic results of the algorithms under the x-relation model are given in Figure 2. The study for the multi-alternative case appears in the full version of this paper [3].

II. The Algorithms

We store D, the set of all N tuples in a relational database table, called the *tuple table*, sorted by the decreasing score order. We store information about the x-tuples in an *x-table*. By using a hash map, given the id of a tuple t, the score and confidence values for all its alternatives can be retrieved efficiently from the x-table in $O(1)$ time.

To process a top-k query, we retrieve tuples in decreasing score order and stop as soon as we are certain that none of the unseen tuples may possibly affect the query result. We define the *scan depth*, denoted by n, to be the minimum number of tuples that have to be retrieved so as to guarantee the correctness of the result. More formally:

*Definition 3 (*Scan depth): Suppose the tuples in an uncertain database $\mathcal{D}$ are $t_1, \ldots, t_N$ in some predefined order. For a U-Top$k$ or U-kRanks query, the *scan depth* n is the minimum n such that the following holds: for any $\mathcal{D}'$ where the first n tuples in $\mathcal{D}'$ under the same ordering criteria are the same as those of $\mathcal{D}$, i.e., $t_1, \ldots, t_n$, the query answer on $\mathcal{D}'$ is the same as that on $\mathcal{D}$.

A. Uncertain Top-k Queries

Define $\mathcal{D}_i$ to be the uncertain database when $\mathcal{D}$ is *restricted* on $D_i = \{t_1, \ldots, t_i\}$, for $i = 1, \ldots, N$, i.e., $\mathcal{D}_i = \{\tau' \mid \tau' = \tau \cap D_i, \tau \in \mathcal{D}\}$. For the database from Figure 1, this means that $\mathcal{D}_1 = \{\tau'_1 = \{t_1\}\}$, $\mathcal{D}_2 = \{\tau'_1 = \{t_1\}, \tau'_2 = \{t_2\}\}$, $\mathcal{D}_3 = \{\tau'_1 = \{t_1\}, \tau'_2 = \{t_2\}, \tau'_3 = \{t_3\}\}$ and $\mathcal{D}_4 = \{\tau'_1 = \{t_1, t_4\}, \tau'_2 = \{t_2\}, \tau'_3 = \{t_3\}\}$. We use $W|\mathcal{D}_i$ to denote a possible world W generated from $\mathcal{D}_i$, with probability $\Pr[W|\mathcal{D}_i]$. For $i \geq k$, let S_i be the most probable world generated from $\mathcal{D}_i$ that consists of k tuples, i.e., $S_i = \arg\max_{|W|=k} \Pr[W|\mathcal{D}_i]$, and let $\rho_i = \Pr[S_i|\mathcal{D}_i]$. Our algorithm follows the following general framework: we read tuples one by one, and progressively compute S_i as i goes from k to N. Finally we take the S_i with the maximum ρ_i as the final answer T^*. The correctness of this general framework is guaranteed by the following lemma.

Lemma 1: $\Pr[\Psi(W|\mathcal{D}) = T^*] = \max\{\rho_i \mid k \leq i \leq N\}$.

Proof: Let $i^* = \max\{i \mid t_i \in T^*\}$. It is clear that $\Pr[\Psi(W|\mathcal{D}) = T^*] = \Pr[\Psi(W|\mathcal{D}_{i^*}) = T^*] = \rho_{i^*}$, so $\Pr[\Psi(W|\mathcal{D}) = T^*] \leq \max\{\rho_i \mid k \leq i \leq N\}$.

On the other hand, consider any T' and let $i' = \max\{i \mid t_i \in T'\}$. By definition $\Pr[\Psi(W|\mathcal{D}) = T^*] \geq \Pr[\Psi(W|\mathcal{D}) = T'] = \rho_{i'}$ for any i'. Thus we have $\Pr[\Psi(W|\mathcal{D}) = T^*] = \max\{\rho_i \mid k \leq i \leq N\}$. ■

Using Lemma 1, instead of computing T^* by Definition 1, i.e., enumerating all the worlds and calculating the maximum aggregated probability, we could simply compute the ρ_i's, and the S_i corresponding to the maximum ρ_i will be T^*. Therefore, the problem boils down to computing S_i and ρ_i for $i = k, k+1, \ldots, N$. In fact, we can stop the process as soon as we are certain that none of the remaining ρ_i's is going to be larger than the maximum ρ_i we have found so far, i.e., as soon as we have read n tuples, where n is the scan depth. However, we still need an efficient algorithm to compute these S_i's and ρ_i's, as well as a method that can tell us if the scan depth is reached or not.

Lemma 2: For a single-alternative database $\mathcal{D}$ and any $k \leq i \leq N$, S_i consists of the k tuples with the largest confidences in D_i, and

$$\rho_i = \prod_{t_j \in S_i} p(t_j) \cdot \prod_{t_j \in D_i \setminus S_i} (1 - p(t_j)).$$

Proof: Since $\Pr[W|\mathcal{D}_i]$ is the product of two factors, the probability that all tuples in W appear and the probability that none of the rest appears, both of which are maximized when W consists of the k largest-confidence tuples. Once we have S_i, ρ_i is immediate. ■

We next characterize the scan depth for this case.

Lemma 3: For a single-alternative uncertain database $\mathcal{D}$ and a U-Top$k$ query, the scan depth is the minimum n such that

$$\max_{1 \leq i \leq n} \rho_i \geq \prod_{1 \leq i \leq n} \max\{p(t_i), 1 - p(t_i)\}. \quad (1)$$

Proof: We first show that when (1) happens, no more tuples need to be fetched. This is because the LHS of (1) is the current best answer we have found after reading n tuples; while the RHS of (1) is an upper bound on $\Pr[W|\mathcal{D}_i]$ for any W, regardless of its cardinality, and any $i > n$.

Next we prove that if (1) does not hold, then we must have not reached the scan depth yet, i.e., the condition is tight. This guarantees that our algorithm will not read more than the necessary n tuples. We first prove the following claim: If we have seen k tuples with confidence $\geq 1/2$, then (1) must hold. Indeed, consider the first time we have seen k such tuples, say after reading t_s. Since the k tuples with the largest confidences in D_s must be those k tuples with confidences $\geq 1/2$, combining with Lemma 2 we have $\max_{1 \leq i \leq s} \rho_i \geq \rho_s = \prod_{1 \leq i \leq s} \max\{p(t_i), 1 - p(t_i)\}$. Furthermore, since the LHS of (1) never decreases and the RHS of (1) never increases, it must still hold when we have read n tuples.

Now, we construct another $\mathcal{D}'$, whose first n tuples are the same as $\mathcal{D}$, while all of its remaining tuples have confidence 1, and argue that we can find a better U-Top$k$ answer from

$\mathcal{D}'$ than the claimed best answer for $\mathcal{D}$ if (1) has not been met yet. Since (1) does not hold, there are $\ell < k$ tuples with confidences $\geq 1/2$ in the first n tuples of $\mathcal{D}$ and $\mathcal{D}'$ as we have just argued. Since all the remaining tuples in $\mathcal{D}'$ have confidence 1, putting together these ℓ seen tuples and the first $k-\ell$ unseen tuples gives us a candidate top-k answer for $\mathcal{D}'$ with probability $\prod_{1\leq i\leq n} \max\{p(t_i), 1-p(t_i)\}$, larger than the current best answer claimed for $\mathcal{D}$. Therefore, by definition we have not reached the scan depth. ■

Using Lemmas 2 and 3, it is easy to obtain an efficient algorithm for processing a U-Top$k$ query. The algorithm reads the tuples one by one, maintains the k largest-confidence tuples seen so far, and computes each ρ_i using Lemma 2. We can use a heap of size k for this purpose, costing $O(\log k)$ time per tuple. Meanwhile, it maintains the RHS of (1) so as to be able to stop immediately after reading n tuples. This can be easily done in constant time per tuple. Therefore we conclude with the following.

Theorem 1: For a single-alternative uncertain database, our algorithm can process a U-Top$k$ query by reading n tuples and spending $O(n \log k)$ time. The space requirement is $O(k)$.

B. Uncertain k-Ranks Queries

In this section we consider U-kRanks queries using a dynamic programming algorithm. Our algorithms are based on the following simple intuition: the probability that a tuple t_i appears at rank j depends only on the event that exactly $j-1$ tuples from the first $i-1$ tuples appear, no matter *which* tuples appear.

Let $\mathcal{D}$ be a single-alternative uncertain database. For $1 \leq j \leq i \leq N$, let $r_{i,j}$ be the probability that a randomly generated world from $\mathcal{D}_i$ has exactly j tuples, i.e., $r_{i,j} = \sum_{|W|=j} \Pr[W|\mathcal{D}_i]$. We also define $r_{0,0} = 1$. It is clear that the probability that t_i ranks the j-th in a randomly generated world from $\mathcal{D}$ is $p(t_i) \cdot r_{i-1,j-1}$. Therefore, the answers to a U-kRanks query on $\mathcal{D}$ are $t_{\chi(j)}$ where

$$\chi(j) = \arg \max_{j\leq i\leq N} \{p(t_i) \cdot r_{i-1,j-1}\}, \tag{2}$$

for $j = 1, \ldots, k$.

We are now left with the task of computing the $r_{i,j}$'s, which are related by the following equation.

$$r_{i,j} = \begin{cases} p(t_i)r_{i-1,j-1} + (1-p(t_i))r_{i-1,j}, & \text{if } i \geq j \geq 0; \\ 1, & \text{if } i = j = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The correctness of (3) is obvious: To get j tuples from $\mathcal{D}_i$, we either choose t_i and $j-1$ tuples from $\mathcal{D}_{i-1}$, or not choose t_i and take all j tuples from $\mathcal{D}_{i-1}$.

Upon reading each tuple t_i, our algorithm computes $r_{i,j}$ using (3) for $j = 0, 1, \ldots, \min\{i, k\}$. It also keeps the current best answers $\chi(j)$ found so far according to (2). Since to compute $r_{i,j}$, only the $r_{i-1,j}$'s are needed, our algorithm only requires $O(k)$ space throughout the computation.

Finally, we have the following characterization of the scan depth n, so that our algorithm can terminate as soon as the answers are known, retrieving only n tuples from the tuple table, which is the minimum possible.

Lemma 4: For a single-alternative uncertain database $\mathcal{D}$ and a U-kRanks query, the scan depth is the minimum n such that the following holds for each $j = 1, \ldots, k$:

$$\max_{j\leq i\leq n} \{p(t_i)r_{i-1,j-1}\} \geq \max_{0\leq \ell\leq j-1} r_{n,\ell}. \tag{4}$$

Proof: Since the LHS of (4) is the current best answer for the tuple at rank j, it is sufficient to prove that, for any $\mathcal{D}'$ whose tuples are $t_1, \ldots, t_n, t'_{n+1}, \ldots, t'_N$, the RHS of (4) is an upper bound on the probability of any t'_i being at rank j for $j = 1, \ldots, k$, and this upper bound is attainable.

First, for any $i > n$, consider the probability of t'_i being at rank j in a randomly generated world from $\mathcal{D}'$. Letting ξ_s be the probability that exactly s tuples from $\{t'_{n+1}, \ldots, t'_{i-1}\}$ appear (define $\xi_0 = 1$ if $i = n+1$), we have

$$\begin{aligned} \Pr[\psi_j(W|\mathcal{D}') = t'_i] &= p(t'_i) \left(\sum_{\ell=0}^{j-1} r_{n,\ell} \cdot \xi_{j-1-\ell} \right) \\ &\leq \sum_{\ell=0}^{j-1} r_{n,\ell} \cdot \xi_{j-1-\ell} \leq \max_{0\leq \ell\leq j-1} r_{n,\ell}, \end{aligned}$$

where the last inequality holds because $\sum_{s=0}^{j-1} \xi_s \leq 1$. Thus, we need to access at most n tuples before we can report the correct answers.

Secondly, we show that for any j, there is a $\mathcal{D}'$ with some unseen tuple that achieves this upper bound. Set $p(t'_{n+1}) = \cdots = p(t'_N) = 1$, and let $\ell^* = \arg\max_{0\leq \ell\leq j-1} r_{n,\ell}$. Consider the tuple $t'_{n+j-\ell^*}$. The probability that it appears at rank j in a random world from $\mathcal{D}'$ is exactly r_{n,ℓ^*}. Therefore, we also need to access at least n tuples to avoid any mistakes. ■

Since we can check the inequality (4) for all $1 \leq j \leq k$ easily in $O(k)$ time per tuple, the theorem below immediately follows.

Theorem 2: For a single-alternative uncertain database, our algorithm can process a U-kRanks query by reading n tuples and spending $O(nk)$ time. The space requirement is $O(k)$.

III. Conclusion

Extensive experimental results and the algorithms for the multi-alternative case could be found in the full version [3].

IV. Acknowledgement

This work was done while Ke Yi and Feifei Li were working at AT&T Labs-Research. Ke Yi is supported in part by Hong Kong Direct Allocation Grant (DAG07/08). Feifei Li and George Kollios were supported in part by the NSF grant IIS-0133825.

References

[1] P. Agrawal, O. Benjelloun, A. Das Sarma, C. Hayworth, S. Nabar, T. Sugihara, and J. Widom, "Trio: A system for data, uncertainty, and lineage," in *VLDB*, 2006.

[2] M. A. Soliman, I. F. Ilyas, and K. C. Chang, "Top-k query processing in uncertain databases," in *ICDE*, 2007.

[3] K. Yi, F. Li, D. Srivastava, and G. Kollios, "Efficient processing of top-k queries in uncertain databases," Florida State University, Tech. Rep., 2008.