D-031

Efficient Top-*k* Dominating Query on Uncertain Database

Xiang Yu　Takeshi Tokuyama†　Jinhee Chun‡

Abstract

We study the problem of ranking queries on uncertain databases where objects are mutually exclusive. We utilize the top-k dominating query with x-Relation data model, and propose novel dominance criteria to return reliable top-k answers. Moreover, we present pruning rules to reduce the computation. The experiments show that our ranking method is more reliable than the uncertain top-k query method. The runtime and space are also theoretically promising.

Keywords: top-k dominating, uncertain database, x-Relation

1. Introduction

The uncertainty of big data generally refers to the inaccuracy or infidelity due to data noise, abnormality and biases, etc.. A top-*k* ranking query is an essential query type on such databases. It returns the k most relevant answers according to the user's preferences. The rank is determined by the score computed with a linear function. An obvious advantage is that users can control the output size using parameter k, but an appropriate ranking function is difficult to specify since it is sensitive to different scales in different dimensions. A skyline query is another important type of query. It returns all objects that are not dominated by others. An object p dominates another object q (denoted by $p \succ q$) in a multi-dimension condition if p is not worse than q on each dimension and strictly better than q on at least one. Although a skyline query does not require users to specify a ranking function, the uncontrolled output size is undesirable. Motivated by the shortcomings of both queries above, we propose the top-*k* that provides a simple and intuitive way to rank tuples. It aims at returning a reliable top-*k* answer with highest dominance scores (e.g., the number of dominated objects) based on the dominance relationship. Our approach has the advantage that it does not need a ranking function and can return exactly k answers where k can be controlled by the user. Moreover, it can satisfy all the ranking properties proposed in [6].

2. Problem Definition

We consider a tuple-level uncertain database with a set of uncertain tuples. The attributes of each tuple t are certain, but the tuple appears with a probability $p(t)$ either taken from a given distribution or uniformly at random. A more complex model that adds mutual exclusion constraints among tuples could naturally arise, which can be specified by a set of generation rules. An example is shown in Table 1. A company has limited resources and some branches. Each branch (denoted by tuple *t*) has an estimated profit score $s(t)$ and a successful probability $p(t)$ based on historical sales referring the probability of obtaining the estimated profit in the next year. The manager needs to allocate the resources into k branches to obtain the maximum profit in the upcoming year. Since branches in the same city generate competition, the k branches should be from different cities. We utilize the *x*-Relation data model presented in the TRIO [1] system. It contains a set of *x*-tuples. Each *x*-tuple τ denoting a generation rule has a set of mutually exclusive tuples. We assume that each tuple appears in a single *x*-tuple, then the *x*-tuples are independent. The tuples in each *x*-tuple are ordered by score and use probability order to break ties. This is arguably a good order whether the number of tuples *N* is known or unknown [2]. The *x*-tuples are ordered by the score of the first tuple. In addition, when the summed probability of the tuples in an *x*-tuple is less than 1, it is possible that no original tuple in the *x*-tuple is present. An example is shown in Table 2. We propose preprocessing computation on the *x*-Relation data model by the following pruning rule: For any two tuples t_1 and $t_2 \in \tau_i$, if $t_1 \succ t_2$, then we prune t_2 from τ_i.

Table 1. An example of tuple-level uncertainty

Tuples	Location	Estimated profit	Probability of success
t_1	Tokyo	125	0.3
t_2	Sendai	110	0.4
t_3	Tokyo	80	0.3
t_4	Tokyo	60	0.4
t_5	Sendai	58	0.5
t_6	Osaka	56	1.0
t_7	Okinawa	49	0.6

Table 2. *x*-Relation data model of example

x-Tuples	Original tuple	Score	Probability
τ_1	t_1	125	0.3
	t_3	80	0.3
	t_4	60	0.4
τ_2	t_2	110	0.4
	t_5	58	0.5
τ_3	t_6	56	1.0
τ_4	t_7	49	0.6

A simple approach is to rank the tuples by the expected score. It is highly dependent on the score that may violate the value invariance [5]. Another expected rank method in [5] can satisfy all properties except faithfulness [6]. Moreover, a U-top*k* query in [4] returns k tuples with maximum aggregated probability. It may ignore a more reliable answer that has a higher score but a slightly lower probability. Considering both of them, we propose a reliable top-*k* dominating query. We define a top-*k* vector (i.e. a set of k original tuples), denoted by *v*, to be a possible query answer. It is associated with a total score s_v and an aggregated probability p_v. When given a tuple-level uncertain dataset, the reliable top-*k* dominating query returns the k tuples that have the highest dominance scores based on the dominance relationship: $v_1 \succ v_2$

† Member of IPSJ, IEICE

‡ Member of IPSJ, IEICE

if vector v_1 dominates v_2 in terms of both total score and aggregated probability.

3. Computation

We define a *Threshold Probability* (denoted by p_{th}) as the aggregated probability of the top-*k* vector that has the highest top-*k* total score. Obviously, the vector with the highest top-*k* total score is the vector that contains the highest scoring tuples of each of the first k *x*-tuples since the *x*-Relation data model ranks by score. It can be computed in $O(1)$ time by:

$$p_{th} = \prod_{1 \le i \le k} p(\tau_i) \quad (1)$$

Where $p(\tau_i)$ is the largest probability of all tuples in τ_i.

Let p_n denote the probability of the current highest aggregated probability top-*k* vector after scanning n *x*-tuples. Then, we define a *Threshold Score* (denoted by s_{th}) as the total score of the vector with highest aggregated probability. It can be computed by the following lemma 1 and lemma 2:

Lemma 1. *In an x-Relation model, let* $q(\tau_i) = 1 - \sum_{t_j \in \tau_i} p(t_j)$ *denote the probability that no tuple in* τ_i *exists in any instance, then the top-k vector that has the highest aggregated probability can be found when the following holds:*

$$p_n \ge \prod_{1 \le i \le n} \max\{p(\tau_i), q(\tau_i)\} \quad (2)$$

Proof. The left hand side of (2) denoting the current highest aggregated probability found after reading n *x*-tuples never decreases, while the right hand side of (2) denoting the upper bound on the probability for any possible instance regardless of its cardinality never increases. Therefore this condition must hold after reading n x-tuples.

Lemma 2. *Let* X_n *be the set of n x-*tuples *retrieved, and let* δ_n^k *denote the set of k x-*tuples *with the largest* $p(\tau_i)/q(\tau_i)$ *ratios in* X_n*, and for any* $x > 0$*, define* $x/0 = \infty$*, then:*

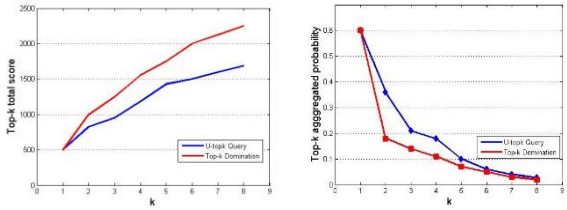
$$p_n = \prod_{\substack{1 \le i \le n \\ \tau_i \in \delta_n^k}} p(\tau_i) \cdot \prod_{\substack{1 \le i \le n \\ \tau_i \in X_n \setminus \delta_n^k}} q(\tau_i) \quad (3)$$

Let $l = |v_i|$ be the number of tuples in a vector, and let $s_{\tau_{n+1}}$ denote the largest scoring tuple in the next *x*-tuple τ_{n+1}. Then $s_{th_l} = s_{th} - (k - l) \cdot s_{\tau_{n+1}}$ is the maximum total score for any incomplete vector ($l < k$), and the aggregated probability never increases. Therefore we have the following vector computation pruning rule:

Lemma 3. *A vector* v_i *will be pruned if the following holds:* $l < k$*, it satisfies* $s_{v_i} < s_{th}$ ($s_{v_i} < s_{th_l}$ *for* $l = k$) and $p_{v_i} < p_{th}$.

4. Experiments

We conducted experiments on a small dataset with some exclusion rules. The results in Figure 1 show that when k increases, the total score of the reliable top-*k* dominating query is larger than that of the U-top*k* query, while the aggregated probabilities are almost the same.



(a) Total score (b) Aggregated probability

Figure 1. Reliability analysis by comparing reliable top-k dominating query and U-topk query based on total score and aggregated probability

5. Conclusion

This work introduces a reliable top-*k* dominating query for uncertain databases under the *x*-Relation data model. Experiments show that the proposed dominance relationship criteria exhibits good reliability. From a theoretical point of view, the algorithm runs in $O(n \log k)$ time and $O(n)$ space. Future work includes doing experiments on larger uncertain databases.

Acknowledgments

Xiang Yu would like to thank Professor Tokuyama and Jinhee for many inspiring discussions and importance advices.

Reference

[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”, Very Large Data Bases, 2006.

[2] K. Yi, F. Li, G. Kollios, D. Srivastava, “Efficient processing of top-k queries in uncertain databases with x-relations”, IEEE Transactions on Knowledge and Data Engineering, 20(12): 1669-1682, 2008.

[3] Y. Tao, X. Xiao, R. Cheng, “Range search on multidimensional uncertain data”, ACM Transactions on Database Systems, 32(3): 15, (2007).

[4] M. A. Soliman, I. F. Ilyas, K. C. Chang, “Top-k query processing in uncertain databases”, IEEE 23rd International Conference on Data Engineering, pages 896-905, 2007.

[5] G. Cormode, F. Li, K. Yi, “Semantics of ranking queries for probabilistic data and expected ranks”, IEEE 25th International Conference on Data Engineering, pages 305-316, 2009.

[6] X. Zhang, J. Chomicki, “Semantics and evaluation of top-k queries in probabilistic databases”, Distributed and Parallel Databases, 26(1): 67-126, 2009.