An Effective Uncertain Data Streams Top-K Query Algorithm

Duan Mingyi* and Lu Yinju*

College of Information and Engineering, Zhongzhou University, Zhengzhou, China

Abstract: Large scale uncertain data streams are produced in many modern applications, such as RFID technology and sensor networks. Top-K query processing is one of the important techniques in the management of uncertain data streams. Existing Top-K queries processing does not consider the score and uncertainty of tuples. This paper first analyzes the uncertain data model and possible world semantic model, and then defines new Top-K queries semantics for uncertain data streams, and finally designs and realizes an effective Top-K queries algorithm on uncertain data streams. This algorithm sorts the score of each tuple and selects the *k* tuples with the highest probabilities to form the set, Top-K queries results. Compared to CSQ and SCSQ algorithms, the experiments show that this algorithm is more practical and effective than the others.

Keywords: Top-K queries, Possible world, Uncertain data streams, Tuple.

1. INTRODUCTION

With the advancement of data mining technology and rapid development of networks technology, more and more data are obtained in domains such as moving object tracing, RFID technology, sensor networks, information extraction and data fusion and so on [1]. How to select information meeting query best in the massive data is an important topic in data management. However, uncertain data streams such as noise, losing value, error and inconsistency, and so on accompany with the massive data from the beginning. The effective query on uncertain data streams is a new topic with the development of modern data processing technology [2-3].

As traditional data management technologies do not apply to uncertain data streams, especially ones in real-time data field. Query based on values, query based on entity, region query and aggregate query, including multi-type Top-K query [4-6], k Nearest Neighbor (kNN) query, are current main researches about uncertain query technology. Top-K query is one of the main applications of database management.

For traditional certain database, Top-K query definition is very clear and it only needs to maintain a buffer of K size to keep Top-K score records. A new arriving record will be compared with K records in the buffer and will replace the low score one. So, comparisons of query result orders are clear and the scores of query results are easy to assign. For example, ticket sales can be Top-K score index of scenic spot database and the number of items can be Top-K score index of three-dimensional warehouse. The meaning is clear.

In the uncertain data streams, uncertainty is expressed as probability in all tuples. So Top-K query becomes unclear and not easy to handle. As a result, it is not applied to base Top-K query on selecting the value function which returns object with the highest value. Two sorting index should be considered: one is the score of query result and the other is the probability of query result. Many ways combining these two exist [7].

Query results vary from definition to definition. Tang Keming, *et al*. proposed an uncertain data streams Top-K query algorithm based on sliding window [8]. Soliman, *et al*. proposed a way based on searching space to handle U-Topk and U-kRanks query [9]. Hua, *et al*. gave a method, constructing advantage set, devoting to PT-k query and a solution is proposed [10].

All the methods above apply to Top-K query in different areas. Different as these query definitions are, they all do not use the related advantage of information retrieval system. People using these queries need to program complicated structured query language, according to the corresponding query syntax. This is a little trouble for an actual user. In addition, these queries are generally high cost. Most of them consider only the probability of database and the scores of data are neglected.

This paper discusses Top-K query based on record vector and proposes RVTK(Record Vector Top-K) algorithm. By the definition of RVTK, a group of Top-K tuples are found, which have the highest scores and probabilities in all possible worlds. This algorithm constructs a summary table based on each record vector, alters summary table to reduce the complexity of Top-K query, and achieves balance between query accuracy and computational cost. Experimental results show that this algorithm has a lower computational cost and better result than other existing ones do.

The main contributions of this paper are as follows:

(1) Gave the Top-K definition and model of uncertain data streams.

*Address correspondence to these authors at the College of Information and Engineering, Zhongzhou University, 6 Yingcai Road, Huiji Area, Zhengzhou Henan Province, China; Tel: +86-371-68229817; Fax: +86-371-68229817; E-mails: duanmingyi@126.com, 15859724@qq.com

(2) Proposed an efficient RVTK query algorithm, which avoids high cost of computing and accords to Top-K retrieval very much.

(3) Proved the algorithm effectiveness by theory and experiment.

2. PROBLEM DESCRIPTION

Definition 1 Uncertain Database tuple: uncertain database is composed of many tuples, which are expressed as: $t = \{x, f(x), p(x)\}$, where x is data item, $f(x)$ is score of data item x, and $p(x)$, which means frequency of occurrence of x in database, is confidence of data item x.

Definition 2 Possible world: uncertain database D is probabilistic database and is based on possible world model. It is composed of a series of instances of possible world. In other word, possible world w is a collection, in uncertain database, of these tuples. Namely, $W = \{w_1, w_2, \cdots, w_n\}$, $P: W \to [0,1]$ is probability distribution, and $\sum_{i=1,\cdots,n} p(w_i) = 1$, $p(w_i) > 0$.

For a possible world w, its probability is defined as:

$$p(w) = \prod_{x\in w} p(x) \prod_{x\notin w} (1 - p(x)). \tag{1}$$

It is obvious that an uncertain database D including n tuples has 2^n possible worlds.

Definition 3 Top-K query: for a positive integer k and a possible world w, if tuple number in w is not less than k, that is $w \geq k$. We sort the tuples in w by their scores calculated by score function f. The first k tuples with high scores is defined as Top-K tuple of w, denoted as $H_k(w)$. Then, we sum probability of possible world corresponding to each Top-K tuple. The one Top-K tuple with maximum probability sum is returned as query result. Let D be an uncertain database and is pw with semantic world. Uncertain Top-K query result T^* on D is defined as:

$$T^* = \arg\max \sum_{w\in pw, H_k(w)=T} p(w). \tag{2}$$

Definition 4 Uncertain data streams: uncertain data streams $S = \{t_1, t_2, \cdots, t_i, \cdots\}$ is composed of a group of sequential tuples. Of which, subscript means timestamp when the tuple is arriving. If there exits an uncertain data stream $S = \{t_1, t_2, \cdots, t_i, \cdots\}$ and a record vector of size L, our Top-K query is to find a proper record vector $S[q] = \{t_{q-L+1}, \cdots t_{q-1}, t_q\}$.

3. TOP-K QUERY ALGORITHM

3.1. Algorithm Framework

First, we propose query algorithm of record vector, which treats record vectors as a static database D with length L. Because processing Top-K query in database D with length L requires enumerating all 2^L possible worlds in PW, this will cost much computation time. So, we propose an effective algorithm to process Top-K query of record vector.

Algorithm Description:

(1) For L tuples $t_1, t_2, \cdots, t_L$ in uncertain database D, sort them by their score $f(t_i)$ in a descending order. We get the new tuple sequence $t_1', t_2', \cdots, t_L'$, and $f(t_i') \geq f(t_{i+1}')$.

(2) In the sequence processed by step(1), the first i tuples is defined as:

$$Q^{(i)} = \{t_1', t_2', \cdots, t_i'\}. \tag{3}$$

(3) Let $H_k^{(i)}$ be the possible world including K tuples. Of which, $i = k+1, k+2, \cdots, L$. These tuples have maximum score in $Q^{(i)}$.

(4) By the Definition 3, the probability $H_k^{(i)}$ appears in $Q^{(i)}$ is expressed as τ_i, where $i = k+1, k+2, \cdots, L$.

(5) Searching for i^* until $\tau_i^* = \max_{k \leq i \leq L} \tau_i$ will get $H_k^{(i^*)}$, the Top-K query result returned in D.

3.2. Algorithm Verification

Expressing Top-K query result of database D as $T_k(D)$, we use $p[T_k(D) = w]$ to denote the Top-K query probability of possible world w in D. $p(w|D)$ represents the appearance probability of possible world w in D.

Lemma 1 Let the set of k tuples be w_k and $w_k = (x_1, x_2, \cdots, x_k)$, and $f(x_1) > f(x_2) > \cdots > f(x_k)$. The scores are sorted in descending order. The set of tuples whose score is not less than $f(x_k)$ is denoted by W_k, and $W_k = \{x \mid x \in D, f(x) \geq f(x_k)\}$. If tuples set $X' = X_k - w_k = \{x_1', x_2', \cdots, x_k'\}$, then we get:

$$p[T_k(D) = w_k] = \prod_{i=1}^{k} p(x_i) \prod_{i=1}^{k} (1 - p(x_i)) . \tag{4}$$

Lemma 1 shows that the Top-K result probability of w_k in D depends only on those tuples whose scores are not less than $f(x_k)$ and all the other tuples are neglected during the computation. Let $Q^{(i)} = \{x_1, x_2, \cdots, x_i\}$ be the set of k tuples with highest scores in database D and let the possible world consisting of these k tuples in $Q^{(i)}$ be $H_k^{(i)}$, and $H_k^{(i)} = \{x_1', x_2', \cdots, x_k'\}$. So, for $x_1', x_2', \cdots, x_k'$, we sort them in descending order by their scores $f(x)$. If $x_k' = x_i$, we say that $H_k^{(i)}$ is the minimum Top-K tuple in $Q^{(i)}$.

By the definition, we get that if $H_k^{(i)}$ is the minimum Top-K tuple in $Q^{(i)}$, it must include tuple x_i, which has minimum score in $Q^{(i)}$.

Lemma 2 Let $H_k^{(i)} = \{x_1', x_2', \cdots, x_k'\}$ be possible world, consisting of k tuples and having maximum probability in $Q^{(i)}$.

The probability of $H_k^{(i)}$ in $Q^{(i)}$ is denoted by $\tau_i = p(H_k^{(i)} \,|\, Q^{(i)})$. If $H_k^{(i)}$ is the minimum Top-K tuple in $Q^{(i)}$, then $\tau_i = p(H_k^{(i)} \,|\, Q^{(i)}) = P[T_k(D) = H_k^{(i)}]$, or else $\tau_i = p(H_k^{(i)} \,|\, Q^{(i)}) < P[T_k(D) = H_k^{(i)}]$.

Proof:

(1) Let $\overline{Q} = Q^{(i)} - H_k^{(i)} = \{\overline{x_1}, \overline{x_2}, \cdots, \overline{x_{i-k}},\}$, then we get:

$$\tau_i = p(H_k^{(i)} \,|\, Q^{(i)}) = \prod_{j=1}^{k} p(x_j^{'}) \prod_{i=1}^{k} (1 - p(x_j^{'})). \tag{5}$$

For $H_k^{(i)}$ is minimum Top-K tuple in $Q^{(i)}$, by Lemma 1, we get:

$$\tau_i = p(H_k^{(i)} \,|\, Q^{(i)}) = P[T_k(D) = H_k^{(i)}]. \tag{6}$$

(2) Among the set $H_k^{(i)} = \{x_1^{'}, x_2^{'}, \cdots, x_k^{'}\}$, let $x_k^{'} = x_{k^{'}}$,of which $x_{k^{'}}$ is a tuple among set $Q^{(i)} = \{x_1, x_2, \cdots, x_i\}$. For $H_k^{(i)}$ is not the minimum Top-K tuple in $Q^{(i)}$, $x_k^{'} < i$ and all of $x_{k+1}^{'}, x_{k+2}^{'}, \cdots, x_i^{'}$ are not in $H_k^{(i)}$.

Let $Q^{(k^{'})} = \{x_1, x_2, \cdots, x_{k^{'}}\}$, for $x_k^{'} = x_{k^{'}}$ and $H_k^{(i)}$ is the minimum Top-K tuple in $Q^{(i)}$, by the conclusion of (1), we get:

$$\tau_{k^{'}} = p(H_k^{(i)} \,|\, Q^{(k^{'})}) = P[T_k(D) = H_k^{(i)}]. \tag{7}$$

For

$$\tau_i = p(H_k^{(i)} \,|\, Q^{(i)}) = p(H_k^{(i)} \,|\, Q^{(k^{'})}) \prod_{j=k^{'}+1}^{i} (1 - p(x_j)) < p(H_k^{(i)} \,|\, Q^{(k^{'})}), \tag{8}$$

we get :

$$\tau_i = p(H_k^{(i)} \,|\, Q^{(i)}) < P[T_k(D) = H_k^{(i)}]. \tag{9}$$

Q.E.D.

Lemma 3 If $T_k(D) = H^*$ is Top-K query result in uncertain database, then $P[T_k(D) = H^*] = \max_i \tau_i$.

Proof:

(1) Because H^* is Top-K query result in uncertain database, for any possible world H with k tuples in uncertain database D, $P[T_k(D) = H^*] \geq P[T_k(D) = H]$. By Lemma 2, for each τ_i and $H_k^{(i)}$, we get:

$$\tau_i \leq P[(T_k(D) = H_k^{(i)}] \leq P[T_k(D) = H^*]. \tag{10}$$

So,

$$P[T_k(D) = H^*] > \max_i \tau_i. \tag{11}$$

(2) Sorting the tuples in D in descending order by their scores, we get $x_1, x_2, \cdots, x_n$, x_I is the tuple in H^* with minimum score. Let W be the set of possible world of K tuples in $Q^{(I)}$. It is obvious that $H^* \in W$. For any tuples set $H \in W$, its probability meets:

$$p(H \,|\, Q^{(I)}) \leq p(H_k^{(I)} \,|\, Q^{(I)}) = \tau_I. \tag{12}$$

Because $H^* \in W$, then we get:

$$p(H^* \,|\, Q^{(I)}) \leq p(H_k^{(I)} \,|\, Q^{(I)}) = \tau_I. \tag{13}$$

Again, because x_I is the tuple with minimum score in H^*, by Lemma 1, we get:

$$P(T_k(D) = H^*) \leq P(H_k^{(I)} \,|\, Q^{(I)}) \leq \tau_I. \tag{14}$$

So,

$$P[T_k(D) = H^*] \leq \max_i \tau_i. \tag{15}$$

(3) From (1) and (2), we get:

$$P[T_k(D) = H^*] = \max_i \tau_i. \tag{16}$$

Q.E.D.

Theorem By the algorithm, we get $\tau_I = \max_i \tau_i$. The Top-K query result in uncertain database D is $T_k(D) = H_k^{(I)}$, and its probability is τ_I that we have gotten.

Proof: Suppose that Top-K query result in uncertain database D is $T_k(D) = H^*$, by Lemma 3, we get:

$$P[T_k(D) = H^*] = \max_i \tau_i. \tag{17}$$

By Lemma 2, for $H_k^{(I)}$ is minimum Top-K tuple set in $Q^{(I)}$, we get:

$$\tau_i = p(H_k^{(I)} \,|\, Q^{(I)}) = P[T_k(D) = H_k^{(I)}]. \tag{18}$$

Since $P[T_k(D) = H^*] = \tau_I = \max_i \tau_i$, hence:

$$P[T_k(D) = H^*] = P[T_k(D) = H^*]. \tag{19}$$

That is $H^* = H_k^{(I)}$ and $T_k(D) = H_k^{(I)}$. So, the probability of Top-K query result is τ_I. Q.E.D.

The theorem shows that in uncertain data streams, the RVTK query algorithm gotten by the algorithm proposed in this paper is theoretically right.

4. EXPERIMENTS

4.1. Experimental Setup

We verify Top-K query performance mainly by experiments. Experimental environment is as follows: Visual Studio 2005, Pentium Dual-Core CPU 2.3GHZ, 4GB RAM, 500GB Hard disk and Windows 7 OS, all the algorithms are written using C language.

For verifying the efficiency of our Top-K query algorithms, in this experiment, we adopt the tourists' traces database of a scenic in Zhangjiajie, Hunan. The database is provided by the tracking and tracing system of tourists' traces built by us [11].

By building visitors touring RFID event collecting network structure and RFID event model and handling mechanism, this database is used to track and trace tourists' traces, improve service quality, prevent shortening tour itinerary and arbitrarily increasing expense and shopping, and fast rescue tourists in danger or missing.

4.2. Experimental Results and Analysis

We test our algorithm in the tourist' traces database. First, for a fix K, compare the efficiency and performance of our algorithm with other ones, such as CSQ or SCSQ algorithms. Then, changing K, analyze and compare the query time and space consumption.

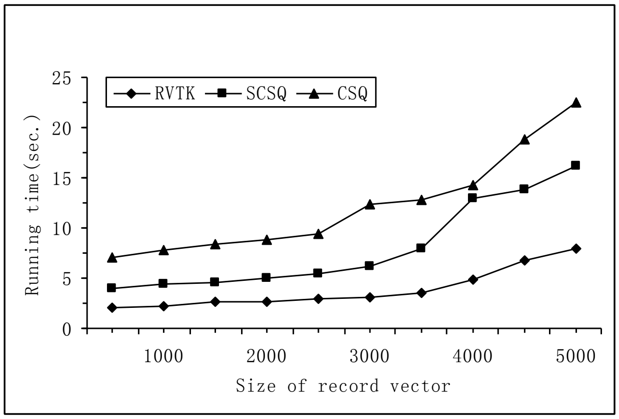

Fig. (1). Computation time of the three algorithms ($k = 50$).

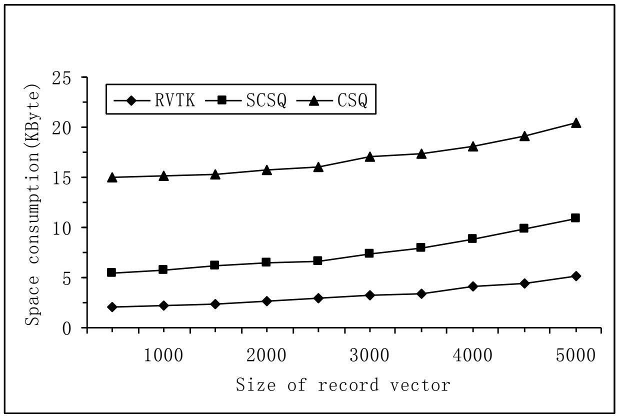

Fig. (2). Space cost of the three algorithms ($k = 50$).

We set $k = 50$, and record vector size L ranging from 500 to 5000. The experimental results of space cost, computational time of single tuple, and all the times are shown by Fig. (**1**) and Fig. (**2**). These experimental results show that compared to CSQ and SCSQ algorithm, our algorithms achieved low time and space consumption.

Fig. (**3**) and (**4**) give the running time and space consumption of querying with our top-K algorithm using top-10, top-20, top-50, and top-100 tuples, respectively. From the figures, we can conclude that our algorithm outperforms the other algorithms in the aspects of time and space.

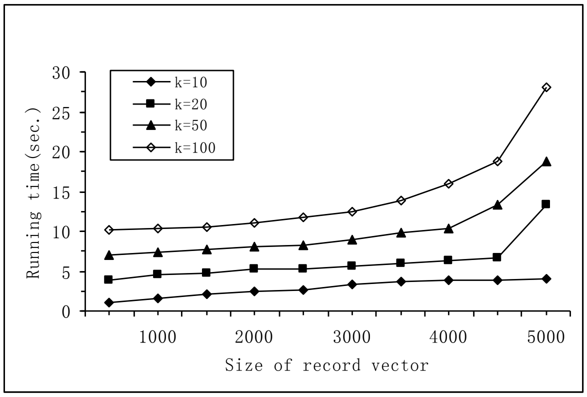

Fig. (3). Running time of algorithm for different k .

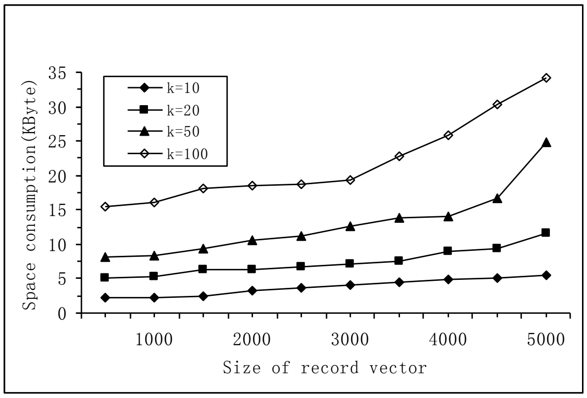

Fig. (4). Space consumption of algorithm for different k .

5. CONCLUSIONS

This paper proposes an effective uncertain data streams Top-K query algorithm, RVTK. This algorithm can facilitate user query and return query result at very low cost. In each record vector, this algorithm selects k tuples with the highest probabilities from different tuples with the highest scores and then it computes the existence probability of Top-K tuple as Top-K query result. Theoretical proof and multi-angle experiments of query algorithm prove the efficiency and practicality of the RVTK query algorithm in this paper.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGMENTS

This work is supported by the Henan province science and technology projects (2102210247).

REFERENCES

[1] A. Silberstein, *A sampling based approach to optimizing top-k queries in sensor networks, Proceedings of the 22nd International Conference on Data Engineering (ICDE)*, Atlanta, GA, USA, 2006, Washington, DC, USA: IEEE Computer Society, pp. 68-80, 2006.

[2] H. Kawashima, *Complex event processing over uncertain data streams, Proceedings of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, Washington, pp. 521-526, 2010.

[3] C.K.S. Leung, *Constrained frequent itemset mining from uncertain data streams. Proceedings of the IEEE 26th international conference on data engineering workshops*, Long Beach, pp. 120-127, 2010.

[4] W. Li, "Top-K query processing techniques on uncertain data", *Journal of Software*, vol. 23, no. 6, pp. 1542-1560, 2012.

[5] X. Li, "Study on Top-k query of uncertainty data in relation database" *Computer Applications and Software*, vol. 29, no. 4, pp. 186-187, 2012.

[6] Z. Zhang, "Top-k keyword query on uncertain database", *Journal of Frontiers of Computer Science and Technology*, vol. 5, no. 9, pp. 781-790, 2011.

[7] X. Zhang and J. Chomicki, "Semantics and evaluation of top-k queries in probabilistic databases", *Distributed and Parallel Databases,* vol. 26, no. 1, pp. 67-126, 2009.

[8] K. Tang, "A Top-K queries algorithm for uncertain data streams based on sliding-window" *Journal of Nanjing University(Natural Sciences)*, vol. 48, no. 3, pp. 351-359, 2012 .

[9] M.A. Soliman, *Top-k query processing in uncertain database[C]. Proceedings of the 23rd IEEE International Conference on Data Engineering*, Istanbul, pp. 896-905, 2007.

[10] M. Hua, *Efficiently answering probabilistic threshold Top-K queries on uncertain data. Proceedings of the 24th IEEE International Conference on Data Engineering*, Washington, pp. 1403-1405, 2008.

[11] Y. Lu, "Study of tracking and tracing system based on RFID" *Coal Technology*, vol. 6, pp. 163-164, 2012.

Received: May 26, 2015 Revised: July 14, 2015 Accepted: August 10, 2015