Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 359137, 14 pages
http://dx.doi.org/10.1155/2015/359137

Research Article

MSSUTop-k: Determining the Minimum Scan Scope for UTop-k Query over Uncertain Data

Zhibin Zhao, Lan Yao, Ge Yu, Yubin Bao, and Zhengbing Ma

Northeastern University, 3-11 Wenhua Road, Shenyang, Liaoning 110819, China

Correspondence should be addressed to Lan Yao; yaolan@mail.neu.edu.cn

Received 5 January 2015; Revised 4 June 2015; Accepted 10 June 2015

Academic Editor: Shaojie Tang

The semantics of UTop-k query is based on the possible world model, and the greatest challenge in processing UTop-k queries is the explosion of possible world space. In this direction, several optimized algorithms have been developed. However, uncertain databases are different in data distributions under different scoring functions, which has significant influence on the performance of the existing optimizing algorithms. In this paper, we propose two novel algorithms, MSSUTop-k and quick MSSUTop-k, for determining the minimum scan scope for UTop-k query processing. This work is important because before UTop-k query processing is started, users hope to know in advance how many and which tuples will be involved in UTop-k query processing. Then, they can make a balance between result precision and processing cost. So, it should be the prerequisite for answering UTop-k queries. MSSUTop-k can achieve accurate results but is relatively more costly in time complexity. Oppositely, quick MSSUTop-k can only achieve approximate results but performs better in time cost. We conduct comprehensive experiments to evaluate the performance of our proposed algorithms and analyze the relationship between the data distribution and the minimum scan scope of UTop-k queries.

1. Introduction

Sensor networks have widespread range of applications these days, such as industrial process monitoring and control, personal health monitoring, environmental monitoring, and moving object tracking. The view of sensor network as a distributed database and the sensor node as a table is being widely accepted by the database community. However, data readings collected from sensors are often inevitably imprecise, and thus they are called uncertain data in academic circles. The uncertainty in the sensed data can arise from multiple sources, including measurement errors due to sensing instruments, transmission delay, and discrete sampling of measurements. Therefore, it is necessary for the sensor database to record the imprecision and also to take it into account when the sensor data is processed. Handling the uncertainty in the data raises great challenges in almost all aspects of sensor data management.

UTop-k query is a crucial application in uncertain database and attracts a lot of attentions in academic circles. UTop-k query answer is a tuple vector with the maximum aggregated probability of being Top-k across all possible worlds. Let us take the uncertain databases in Table 1 as an example. Table 1(a) is a database storing the velocities of vehicles measured by radar. Each reading is associated with a confidence level associated with the corresponding measuring or statistic error. Exclusiveness rules are prior knowledge and are derived from the specific application. In this example, event t_2 is exclusive to t_3 because the car Y-245 cannot be at two different speeds at the same time. Generally, researchers employ possible world (PW) model to describe uncertain database. All possible instances derived from the corresponding uncertain database compose the possible world space. Table 1(b) is the possible worlds space of Table 1(a).

Given an uncertain database as Table 1(a), according to the definition of UTop-k [1], the UTop-2 answer is $\{t_2, t_1\}$. The reason is that the combination of $\{t_2, t_1\}$, as the top-2 answer, appears in four possible worlds: PW_1, PW_2, PW_3, and PW_4, and aggregated probability is $Pr(PW_1) + Pr(PW_2) +$

Table 1: Example of uncertain databases and UTop-k results.

(a) Uncertain database 1

ID	Time	Radar	Plate	Speed	Conf.
t_1	1145	L_1	X-123	115	1.0
t_2	1150	L_2	Y-245	120	0.7
t_3	1150	L_3	Y-245	80	0.3
t_4	1210	L_4	W-541	90	0.4
t_5	1210	L_5	W-541	110	0.6
t_6	1215	L_6	L-105	105	1.0
t_7	**1230**	**L_7**	**L-048**	**40**	**0.5**
t_8	1230	L_8	L-048	55	0.5

Exclusiveness rules: $t_2 \oplus t_3, t_4 \oplus t_5, t_7 \oplus t_8$

(b) Utop-k answer based on uncertain database 1

Possible World	Top-2	Top-3	Conf.
$PW_1 = \{t_1, t_2, t_4, t_6, t_7\}$	$\{t_2, t_1\}$	$\{t_2, t_1, t_6\}$	0.14
$PW_2 = \{t_1, t_2, t_4, t_6, t_8\}$	$\{t_2, t_1\}$	$\{t_2, t_1, t_6\}$	0.14
$PW_3 = \{t_1, t_2, t_5, t_6, t_7\}$	$\{t_2, t_1\}$	$\{t_2, t_1, t_5\}$	0.21
$PW_4 = \{t_1, t_2, t_5, t_6, t_8\}$	$\{t_2, t_1\}$	$\{t_2, t_1, t_5\}$	0.21
$PW_5 = \{t_1, t_3, t_4, t_6, t_7\}$	$\{t_1, t_6\}$	$\{t_1, t_6, t_4\}$	0.06
$PW_6 = \{t_1, t_3, t_4, t_6, t_8\}$	$\{t_1, t_6\}$	$\{t_1, t_6, t_4\}$	0.06
$PW_7 = \{t_1, t_3, t_5, t_6, t_7\}$	$\{t_1, t_5\}$	$\{t_1, t_5, t_6\}$	0.09
$PW_8 = \{t_1, t_3, t_5, t_6, t_8\}$	$\{t_1, t_5\}$	$\{t_1, t_5, t_6\}$	0.09

UTop-2 = $\langle\{t_2, t_1\}, 0.7\rangle$; UTop-3 = $\langle\{t_2, t_1, t_5\}, 0.42\rangle$

(c) Uncertain database 2

ID	Time	Radar	Plate	Speed	Conf.
t_1	1145	L_1	X-123	115	1.0
t_2	1150	L_2	Y-245	120	0.7
t_3	1150	L_3	Y-245	80	0.3
t_4	1210	L_4	W-541	90	0.4
t_5	1210	L_5	W-541	110	0.6
t_6	1215	L_6	L-105	105	1.0
t_7	**1230**	**L_7**	**L-048**	**117**	**0.5**
t_8	1230	L_8	L-048	55	0.5

Exclusiveness rules: $t_2 \oplus t_3, t_4 \oplus t_5, t_7 \oplus t_8$

(d) Utop-k answer based on uncertain database 2

Possible World	Top-2	Top-3	Conf.
$PW_1 = \{t_1, t_2, t_4, t_6, t_7\}$	$\{t_2, t_7\}$	$\{t_2, t_7, t_1\}$	0.14
$PW_2 = \{t_1, t_2, t_4, t_6, t_8\}$	$\{t_2, t_1\}$	$\{t_2, t_1, t_6\}$	0.14
$PW_3 = \{t_1, t_2, t_5, t_6, t_7\}$	$\{t_2, t_7\}$	$\{t_2, t_7, t_1\}$	0.21
$PW_4 = \{t_1, t_2, t_5, t_6, t_8\}$	$\{t_2, t_1\}$	$\{t_2, t_1, t_5\}$	0.21
$PW_5 = \{t_1, t_3, t_4, t_6, t_7\}$	$\{t_7, t_1\}$	$\{t_7, t_1, t_6\}$	0.06
$PW_6 = \{t_1, t_3, t_4, t_6, t_8\}$	$\{t_1, t_6\}$	$\{t_1, t_6, t_4\}$	0.06
$PW_7 = \{t_1, t_3, t_5, t_6, t_7\}$	$\{t_7, t_1\}$	$\{t_7, t_1, t_6\}$	0.09
$PW_8 = \{t_1, t_3, t_5, t_6, t_8\}$	$\{t_1, t_5\}$	$\{t_1, t_5, t_6\}$	0.09

UTop-2 = $\langle\{t_2, t_1\}/\{t_2, t_7\}, 0.35\rangle$; UTop-3 = $\langle\{t_2, t_7, t_1\}, 0.35\rangle$

$\Pr(PW_3) + \Pr(PW_4) = 0.7$. This probability is larger than that of $\{t_1, t_6\}$ and $\{t_1, t_5\}$, which are the top-2 answers in the other four possible worlds and with the probability of 0.12 and 0.18, respectively. Similarly, the UTop-3 answer is $\{t_2, t_1, t_5\}$ with probability 0.42. In Table 1(a), there are totally 8 tuples in the uncertain database. According to the exclusive rules, the possible world space is composed of 8 possible world instances. Theoretically, given an uncertain dataset containing M groups of mutually exclusive tuples, the cardinality of the possible world space will be at least 2^M. It implies that the possible world space grows quite faster than the uncertain dataset itself. This poses a great challenge on UTop-k processing.

Basically, we can use the naive algorithm to process UTop-k queries, in which the possible world space is comletely produced and then the aggregated probability of each UTop-k answer can be obtained. However, the naive algorithm is prohibitively expensive in space and time cost. Actually, in many cases we can achieve the Utop-k answers by reading part of the uncertain dataset. Let us examine the possible world space in Table 1(b). The UTop-2 answer candidates are $\{t_2, t_1\}$, $\{t_1, t_6\}$, and $\{t_1, t_5\}$. The UTop-3 answer candidates are $\{t_2, t_1, t_6\}$, $\{t_2, t_1, t_5\}$, $\{t_1, t_6, t_4\}$, and $\{t_1, t_5, t_6\}$. The tuples t_7 and t_8 in Table 1(a) do not appear either in any of UTop-2 answer candidates or in that of UTop-3 answer candidates. In other words, the subset $\{t_7, t_8\}$, as a group containing two mutually exclusive tuples, only enlarges the possible world space but has no contrition to the final UTop-k answer. So, if we remove the subset of $\{t_7, t_8\}$ from the original uncertain dataset, that is, we set $\{t_1, t_2, t_3, t_4, t_5, t_6\}$ as the minimum scan scope for UTop-k processing, we can obtain accurate UTop-2 and UTop-3 answers while only four possible worlds are generated. Generally, if we eliminate a group with l mutually exclusive tuples, the possible world space will decrease l times. However, in some other cases, the entire uncertain dataset should be scanned for processing UTop-k queries. Let us take Table 1(c) as an example. It is just a little different from Table 1(a) in the value of t_7. Table 1(d) is the possible world space and UTop-2 and UTop-3 answers for Table 1(c). In this example, all the tuples must be considered. From the two examples in Table 1, we can see that, given an uncertain dataset, a user-predefined scoring function, and a parameter k, the number of tuples that are necessarily involved in processing UTop-k queries is different. It should be reduced as much as possible for its great influence on processing time.

In this paper, we propose two different methods to determine scan scope of tuples for UTop-k query processing. According to the minimum scan scope for accurate result, users can make a balance between the result precision with processing time cost. Towards this end, our contributions are summarized as follows:

(i) We propose our basic MSS4UTop-k algorithm for determining the minimum scan scope when UTop-k queries are handled. MSS4UTop-k can obtain the exact minimum scan scope.

(ii) In order to promote the efficiency in scan scope determination, we study some special cases in uncertain dataset and propose the algorithm of Quick

MSS4UTop-k. It enlarges the minimum scan scope but can perform better than MSS4UTop-k in time cost.

(iii) We conducted an extensive experimental study on real uncertain dataset to test the performance of our algorithms. At the same time, we analyze the relationship between the data distribution and the minimum scan scope.

2. Problem Definitions

UTop-k semantics is based on the model of possible worlds [1]. Assume that there is a user-specified scoring function $\mathscr{F}$, under which the tuples in an uncertain database can be sorted. UTop-k query answer is a tuple vector with the maximum aggregated probability of being top-k across all possible worlds.

Definition 1 (UTop-k query). Let $\mathscr{D} = \{t_1, t_2, \ldots, t_N\}$ be an uncertain database with the possible world space $\mathscr{PW} = \{\mathrm{PW}^1, \mathrm{PW}^2, \ldots, \mathrm{PW}^n\}$. Let $\mathscr{T} = \{T^1, T^2, \ldots, T^m\}$ be a set of k-length tuple vectors, where for each $T^i \in \mathscr{T}$ (1) tuples of T^i are ordered according to the scoring function $\mathscr{F}$ and (2) T^i is the top-k answer for a nonempty set of possible worlds $\mathrm{PW}(T^i) \subseteq \mathscr{PW}$. U-Top$k$ query over $\mathscr{D}$, based on $\mathscr{F}$, return $T^* \in \mathscr{T}$, where $T^* = \mathrm{argmax}_{T^i \in \mathscr{T}}(\sum \Pr(\mathrm{PW}(T^i)))$.

Definition 2 (X-tuple bucket B). X-tuple [2] bucket consists of one or more alternatives, where each alternative is a regular tuple exclusive to the others according to the exclusiveness rules derived from applications. Therefore, given an uncertain database $\mathscr{D} = \{t_1, t_2, \ldots, t_N\}$, the X-tuple bucket B is a subset of $\mathscr{D}$, that is, $B \subseteq \mathscr{D}$, and $B = \{t \mid (\forall t' \in B, t \oplus t') \wedge (\forall t' \notin B, \neg(t \oplus t'))\}$.

Tuples in uncertain dataset D can be divided into two categories: (1) tuples with probability 1, which is called deterministic tuples, and (2) tuples with probability less than 1. Tuples in the first category appear in all possible worlds deterministically, while those in the second category appear in some of possible worlds at their own probability. Furthermore, we can divide the tuples into subsets according to the exclusiveness rules. Each deterministic tuple compose a subset itself, and the mutually exclusive tuples are organized into one X-tuple bucket. Then, the uncertain dataset can also be denoted as $\mathscr{D} = B_1 \cup B_2 \cup, \ldots, \cup B_M$, where each B_i is X-tuple bucket.

Definition 3 (UTop-k minimum scan scope $\mathscr{D}^s$). Given an uncertain database $\mathscr{D} = \{t_1, t_2, \ldots, t_N\}$, the minimum scan scope $\mathscr{D}^s$ for UTop-k processing is a subset of $\mathscr{D}$ and satisfies the following two criteria: (1) $\mathscr{D}^s$ includes the minimum X-tuple buckets and (2) $\mathscr{D}^s$-based UTop-k answer is the same as $\mathscr{D}$-based UTop-k answer.

Obviously, $\mathscr{D}^s$ is related to k and data distribution of uncertain dataset. The word "minimum" here has two meanings: (1) tuples in $\mathscr{D} - \mathscr{D}^s$ will not enter any Top-k candidate set across the possible world space based on $\mathscr{D}$; (2) $\mathscr{D}^s$ includes sufficient tuples to avoid possible errors.

In this paper, we aim at determining $\mathscr{D}^s$. For convenience of later discussion, we summarize our notations in Notations.

3. Related Work

Top-k queries in deterministic database is always a hot topic in academia, and several efficient methods for optimizing Top-k queries have been proposed [3–5]. However, in recent years it is noticed that physical data is usually uncertain and/or fuzzy [6–11]. The marriage of Top-k and uncertain data starts a novel research issue: Top-k queries in uncertain database. There are several semantics of Top-k queries in uncertain database: UTop-k [1], U-kRanks [1], PT-k [12], PKTop-k [13], Expected Ranks [14, 15], and other recent research works in [16–20].

In this paper, we focus on the semantics of UTop-k. The work in [1] is the first to introduce the definition of UTop-k. It proposed the OptUTop-k framework for UTop-k processing as well. Its basic idea is based on the following two assumptions: (1) tuples in uncertain database is accessed one by one sequentially; that is, no random access is allowed; (2) the global exclusiveness rules are unknown in advance. OptUTop-k maintains a priority state queue which is ordered on probability. Then, it reads tuples one by one sequentially. Each time when a new tuple arrives, the top state in the priority state queue will be extended into two new states, with the newly seen tuple and without the newly seen tuple. By searching all possible states, OptUTop-k can obtain the most probable top-k answers.

Another important work in UTop-k processing is introduced in [21]. The basic idea in [21] is based on the following two assumptions: (1) tuples in uncertain dataset are ordered according to the scoring function; (2) the global exclusiveness rules are known in advance. Based on these two assumptions, the optimizing framework reads the tuples one by one in sequence of scores. Each time when a new tuple arrives, the possible world space will be produced based on all seen tuples. This procedure repeats until the scan depth is reached. Then, the Top-k candidate set in the possible worlds with the highest probability is the final UTop-k answer.

Our work in this paper is different from the two works above. Firstly, they are different in preconditions. Our work is based on the following two assumptions: (1) the global exclusiveness rules are known and (2) the whole uncertain dataset can be traversed in advance; that is, N is known. The justification for our first assumption is that the exclusiveness rules come from applications, and with the help of domain experts we can translate this prior knowledge into user-defined constraints in database at the very beginning. The justification for our second assumption is that uncertain data is usually stored in RDBMS, in which traversing a table is a common operation. Secondly, they are different in goals. The two works above aim at optimizing UTop-k processing. However, we view our work as a prerequisite step for UTop-k processing. We emphasize that before we start any optimized algorithm, we must determine the necessary scan scope in uncertain database for processing

UTop-k query; that is, how many and which tuples are necessarily involved for query processing? It is important because in some cases all the uncertain tuples may be involved in answering UTop-k queries. In such scenarios, any effort seeking for precision results will lead to failure, and the approximate algorithms are the best (or say the only) choice.

Scoring function is another interesting research point in rank queries. Soliman et al. [22] notice that in many cases users cannot precisely specify their scoring functions, which means that in such scenarios the scoring function is uncertain or incomplete. This work is different from ours for two aspects: (1) they are different in data model. In the work of [22], data is determined, while the scoring function it is uncertain. Therefore, they model their data with traditional database (deterministic database). Oppositely, in our work, we assume that data gathered from application field is with uncertainty, while the scoring function is determined. Researchers are inclined to make use of probability database and possible world model to describe uncertain dataset. (2) They are different in query semantics. Soliman et al. set their focus on the "uncertain/incomplete scoring functions" in Top-k queries. The question is that "Can we adopt a score function with weight ranges and partially specified weight for different data sources to capture the preferences of users so as to give them a personalized Top-k result?". Furthermore, they analyze the sensitivity of the computed order with respect to changes in weights. In our work, we aim at the semantic of UTop-k over uncertain data. The most challenging problem in processing UTop-k queries is the explosion of possible world space, which makes it unfeasible for its quite expensive cost in processing time. However, in our paper we demonstrate that in many cases not all tuples in uncertain database are necessarily involved in answering UTop-k queries. So, we set our goal to conclude that "which is the minimum necessary scan scope of tuples in uncertain data for UTop-k query processing?". In summary, although these two works are both about "uncertainty" and "Top-k queries," they are essentially different.

Besides, query result evaluation and data cleaning are always an attractive research problem in the field of uncertain data management. Given the data uncertainty, a query answer is inherently inexact. Paper [23] puts forward the measurement method of the ambiguity of the query results. If users are not satisfied with the quality of query answer, a cleaning process will be launched. Ideally, all X-tuples should be cleaned. However, this cleaning process is limited by power resource, bandwidth, budget, successfulness, and so on, which makes it an optimization problem for users. Xu et al. [24] set their focus on the problem of automatically selecting an extractive summary from entire set of objects as its representatives. Practically, objects may have multiple uncertain attributes. So, paper [24] proposes a general framework that models the information contained in objects and optimizes a probabilistic coverage property of the summary. Although all these works are based on uncertain data, their research point is not about the semantic and processing of UTop-k queries.

4. MSS4UTop-k Framework

In this section, we will introduce our algorithms to determine the minimum scan scope for UTop-k query processing in uncertain databases. In most cases, k is far less than N. It implies that some of, or maybe most of, tuples have no influence on any Top-k candidates. According to this basic idea, we believe that some of tuples in uncertain dataset can be safely pruned when UTop-k query is processed. Then, the remaining part is the minimum scan scope. The challenge is that how we can distinguish those "useless" tuples.

4.1. Phrases of MSS4UTop-k. MSS4UTop-k has three phrases: first, we traverse the uncertain dataset, group the tuples in $\mathscr{D}$ according to exclusiveness rules, and choose one representative tuple for each X-tuple group. We sort these representative tuples under the scoring function $\mathscr{F}$ and select top-k representative tuples to compose our representative tuple set T^r. Next, we extend the representative set. In some possible worlds, some tuples in the representative set may be absent from the top-k candidates and replaced by others. So, we extend the representative set to make sure that all the tuples that have chance to enter any top-k result set, no matter in which possible world, must be included. Finally, We determine the minimum scan scope based on the extended representative tuple set.

4.1.1. Initializing the Representative Tuple Set T^r. First, as we mentioned in Section 2, we assign all the tuples in $\mathscr{D}$ into X-tuple buckets; that is, $\mathscr{D} = B_1 \cup B_2 \cup, \ldots, \cup B_M$. Our distribution rules are (1) each deterministic tuple has one bucket by itself and (2) tuples that conflict with each other under exclusiveness rules are put into the same bucket. Next, for each bucket, we select the top-1 tuple under the scoring function $\mathscr{F}$ as the representative tuple. We use $t_{B_i}^{\max}$ to denote the representative tuple for the bucket B_i. Obviously, for $\forall t_j \in B_i$, $\mathscr{F}(t_{B_i}^{\max}) > F(t_j)$. We sort all the representative tuples according to their scores under $\mathscr{F}$. Without loss of generality, the sorted representative tuples can be denoted as fully ordered vector $T' = \langle t_{B_1}^{\max} \preceq_{\mathscr{F}} t_{B_2}^{\max} \preceq_{\mathscr{F}} \cdots \preceq_{\mathscr{F}} t_{B_i}^{\max} \cdots \preceq_{\mathscr{F}} t_{B_M}^{\max} \rangle$, and the corresponding buckets behind also have their order: $B_1 \preceq_{\mathscr{F}} B_2 \preceq_{\mathscr{F}} \cdots \preceq_{\mathscr{F}} B_i \cdots \preceq_{\mathscr{F}} B_M$. For simplicity of discussion, we assume that all the scores of the tuples in T' are distinct. Figure 1 illustrates how T' is generated under uncertain dataset in Table 1(c).

When a UTop-k query is initiated, we first locate the kth element in T'. Then, the elements from $t_{B_1}^{\max}$ to $t_{B_k}^{\max}$ compose our initial representative tuple set $T^r = \langle t_{B_1}^{\max}, t_{B_2}^{\max}, \ldots, t_{B_k}^{\max} \rangle$. Obviously, $T^r \subseteq T'$. Next, we need to extend T^r so that it can cover all the X-tuple buckets that contain the tuples with chance to enter any T^i in $\mathscr{T}$. We define a lower bound tuple $t_{\mathrm{lb}}^{\max}$. It is always assigned with the element next to the last element of T^r in T'. Undoubtedly, $t_{\mathrm{lb}}^{\max}$ is corresponding to the X-tuple bucket B_{lb}, and initially $t_{\mathrm{lb}}^{\max} = t_{B_{k+1}}^{\max}$; that is, $\mathrm{lb} = k + 1$ because $t_{B_k}^{\max}$ is the last element of T^r in the very beginning.

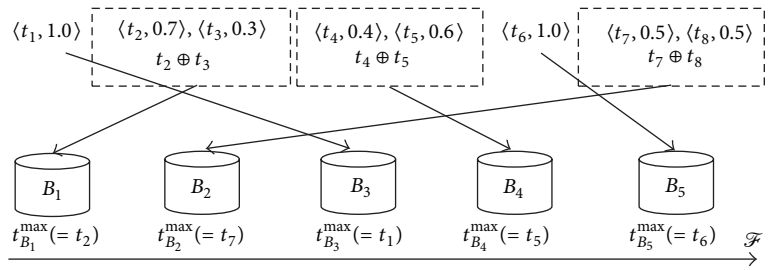


FIGURE 1: T' under uncertain dataset in Table 1(c).

4.1.2. Extending the Representative Tuple Set T^r. We start to traverse T^r from the beginning to the end. Before we reach the element $t^{\max}_{B_k}$, we may meet the following two cases.

Case 1. We meet a deterministic tuple $t^{\max}_{B_i}$ $(i \le k)$.

Theorem 4. *Assume that $t^{\max}_{B_i}$ is a deterministic tuple in T^r. Then, one can conclude the following: (1) $\forall T^i$ $(T^i \in \mathcal{T})$, $t^{\max}_{B_i} \in T^i$, and (2) $t^{\max}_{B_i} \in T^*$.*

Proof. Since $t^{\max}_{B_i} \in T^r \subseteq T'$ and T' is ordered under scoring function $\mathcal{F}$, it can be inferred that $\mathcal{F}(t^{\max}_{B_i}) > \mathcal{F}(t^{\max}_{\text{lb}}) > \cdots > \mathcal{F}(t^{\max}_{B_M})$. Moreover, according to the definition, $\mathcal{F}(t^{\max}_{B})$ is the maximum in any X-tuple bucket B; then we can conclude that $\mathcal{F}(t^{\max}_{B_i})$ is larger than any $\mathcal{F}(t)$, where $t \in B_{j\ (k+1\le j\le M)}$. This means that $t^{\max}_{B_i}$ must appear in the top-k answer sets of all possible worlds. Furthermore, its rank under scoring function $\mathcal{F}$ is at most i $(i \le k)$. So, (1) $\forall T^i$ $(T^i \in \mathcal{T})$, $t^{\max}_{B_i} \in T^i$, and (2) $t^{\max}_{B_i} \in T^*$ are proved. □

According to Theorem 4, a deterministic tuple $t^{\max}_{B_i}$ $(i \le k)$ must have a position in T^*. So, in this scenario, we keep T^r unextended. This is illustrated in Figure 2.

Case 2. We meet a tuple with probability less than 1; that is, $\Pr(t^{\max}_{B_i}) < 1$ $(i \le k)$.

In this case, the bucket B_i contains more than one tuples, and $\sum_{t\in B_i}(\Pr(t) = 1)$. Essentially, there are two more specific cases here: (1) $\min(\mathcal{F}_{t\in B_i}(t)) > \mathcal{F}(t^{\max}_{\text{lb}})$ and (2) $\exists t$, $\mathcal{F}_{t\in B_i}(t) \le \mathcal{F}(t^{\max}_{\text{lb}})$. We discuss these two scenarios in detail, respectively.

Case 2.1 $(\min(\mathcal{F}_{t\in B_i}(t)) \ge t^{\max}_{\text{lb}})$. This case is quite similar to Case 1. We call it the best situation. Since $\min(\mathcal{F}_{t\in B_i}(t)) \ge t^{\max}_{\text{lb}}$, we can conclude that $\min(\mathcal{F}_{t\in B_i}(t)) > \mathcal{F}_{t\in B_{j\ (k+1\le j\le M)}}(t)$. It implies that B_i must devote one tuple to the top-k candidate answer set in any of the possible worlds. Similarly, in this scenario, we still keep T^r unextended, as depicted in Figure 2.

Case 2.2 $(\exists t, \mathcal{F}_{t\in B_i}(t) \le \mathcal{F}(t^{\max}_{\text{lb}}))$. Assume that there are l elements in B_i, including $t^{\max}_{B_i}$; that is, $B_i = \{\langle t^1_{B_i}, \Pr(t^1_{B_i})\rangle, \ldots, \langle t^{\max}_{B_i}, \Pr(t^{\max}_{B_i})\rangle, \ldots, \langle t^l_{B_i}, \Pr(t^l_{B_i})\rangle\}$. We consider the worst situation $\forall t$ $(t \ne t^{\max}_{B_i})$, $\mathcal{F}_{t\in B_i}(t) \le \mathcal{F}(t^{\max}_{\text{lb}})$. Under this worst assumption, $t^{\max}_{B_i}$ will appear in n/l possible worlds, and it undoubtedly must take a position in top-k set in the n/l possible worlds. However, in the other $n(1 - 1/l)$ possible worlds, tuples from B_i are kicked out of the top-k set and replaced by tuples from other buckets such as B_{lb}. So, we have to extend T^r in order to make sure that it can cover all the representative tuples whose corresponding buckets contain the tuples with chance to enter top-k candidate sets. Extension of T^r as well as its start and stop principles are as follows.

(i) *Extension Start Principle.* If $\exists t \in B_i$, $\mathcal{F}(t) \le \mathcal{F}(t^{\max}_{\text{lb}})$, T^r extension starts.

(ii) *Extension Principle.* $T^r = T^r \cup \{t^{\max}_{\text{lb}}\}$, and $t^{\max}_{\text{lb}}$ will be placed to the tail of T^r.

(iii) *Extension Stop Principle 1.* If $t^{\max}_{\text{lb}}$ is a deterministic tuple, T^r extension stops.

(iv) *Extension Stop Principle 2.* If, $\forall t \in B_{\text{lb}}$, $\mathcal{F}(t) > \mathcal{F}(t^{\max}_{B_{\text{lb}+1}})$, T^r extension stops.

(v) *Extension Stop Principle 3.* If, $\forall t \in B_i$, $\mathcal{F}(t) > \mathcal{F}(t^{\max}_{B_{\text{lb}+1}})$, T^r extension stops.

(vi) *Extension Stop Principle 4.* If the end of T' is reached, that is, $\text{lb} = M$, T^r extension stops.

Besides Cases 2.1 and 2.2, there are still another situation which is between the best one and the worst one, more than one tuple whose score is larger than $\mathcal{F}(t^{\max}_{\text{lb}})$, and others are the opposite. We call it the intermediate situation. This intermediate situation should be handled in the same way as the worst one because, as we mentioned above, we must guarantee that all the tuples with chance to enter top-k candidate set have their bucket representative in T^r. So, the extension of T^r is necessary in this intermediate case.

In the above cases, we listed all of the possible situations of $t^{\max}_{B_i}$. After $t^{\max}_{B_i}$ is handled, we move to the next element in T^r. We keep traversing T^r until the element $t^{\max}_{B_k}$ is handled. T^r becomes larger and larger in this process. Figure 3 illustrates how T^r is extended with $t^{\max}_{\text{lb}}$.

4.1.3. Determine the Minimum Scan Scope $\mathcal{D}^s$. After being extended, T^r is $\langle t^{\max}_{B_1}, \ldots, t^{\max}_{B_k}, \ldots, t^{\max}_{B_j}\rangle$. According to the Extension Principle and the Extension Stop Principles, we can conclude that the buckets corresponding to the tuples in the current T^r, that is, $B_1, \ldots, B_k, \ldots, B_j$, contain all

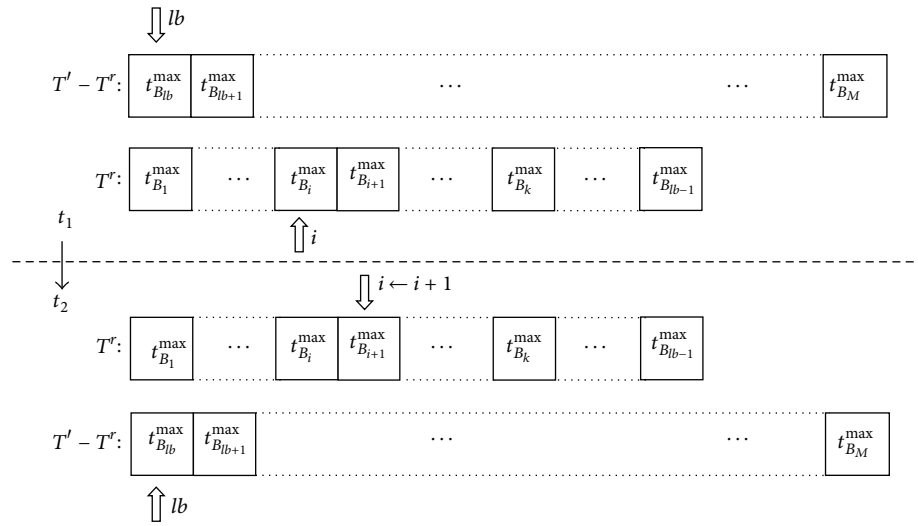


FIGURE 2: T^r keeps unchanged if $\Pr(t^{\max}_{B_i}) = 1$ or $\min(\mathscr{F}_{t\in B_i}(t)) \geq \mathscr{F}(t^{\max}_{\text{lb}})$.

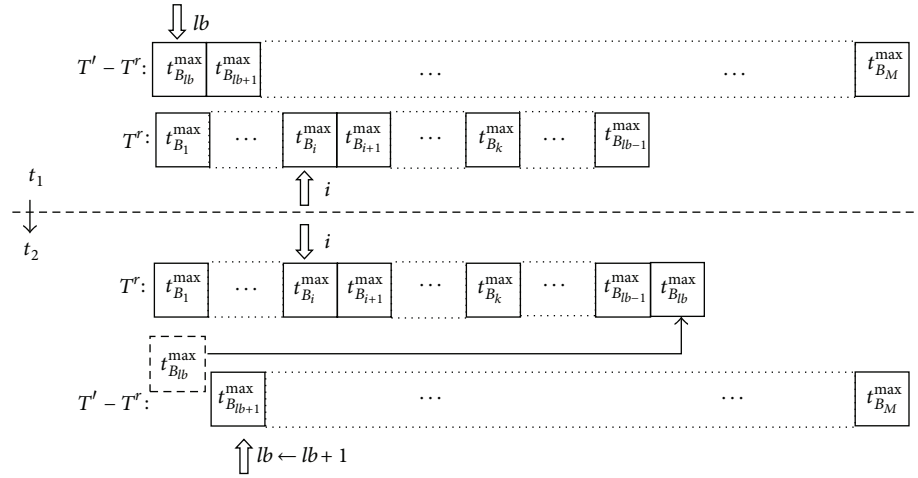


FIGURE 3: The extension of T^r: $T^r = T^r \cup t_{B^{\max}_{\text{lb}}}$.

the tuples that have chance to enter the Top-k result sets of some possible worlds. We combine all the tuples in $B_1, \ldots, B_k, \ldots, B_j$, and the minimum scan scope is obtained; that is, $\mathscr{D}^s = \bigcup_{t^{\max}_{B_i}\in T^r} B_i$. Algorithm 1 describes how $\mathscr{D}^s$ is generated in detail.

4.1.4. Correctness. In this section, we will prove that the $\mathscr{D}^s$-based UTop-k result equals the $\mathscr{D}$-based UTop-k result.

Theorem 5. *Assume that the original uncertain dataset is $\mathscr{D}$. $\mathscr{D}^s$ is the minimum scan scope for UTop-k queries on $\mathscr{D}$. $\mathscr{PW}_{\mathscr{D}^s}$ is the $\mathscr{D}^s$-based possible world space, and $\mathscr{T}_{\mathscr{D}^s}$ is all the top-k candidates derived from $\mathscr{PW}_{\mathscr{D}^s}$. $T^*_{\mathscr{D}^s}$ is the UTop-k answer based on $\mathscr{PW}_{\mathscr{D}^s}$. Then, $T^* = T^*_{\mathscr{D}^s}$.*

Proof. $\mathscr{PW} = \{\text{PW}^1, \text{PW}^2, \ldots, \text{PW}^n\}$ is the $\mathscr{D}$-base possible world space, and $\mathscr{T}$ is k-length tuple vectors, where each $T^i \in \mathscr{T}$ is a possible top-k solution derived from $\mathscr{PW}$. Definitely, if the naive approach is adopted, we can get the correct UTop-k answer $T^* = \text{argmax}_{T^i\in\mathscr{T}}(\sum \Pr(\text{PW}(T^i)))$. If we use the proposed MSS4UTop-k, we can get $\mathscr{D}^s = \bigcup_{i=1}^{j\ (j\leq M)} B_i$. It implies that any tuples from the set $B_{j+1}, B_{j+2}, \ldots, B_M$ have no chance to enter any T^i ($T^i \in \mathscr{T}$).

Case 1 ($j = M$). Consider

$$\begin{aligned} j = M &\Longrightarrow \\ \mathscr{D}^s = \mathscr{D} &\Longrightarrow \\ T^* = T^*_{\mathscr{D}^s}&. \end{aligned} \tag{1}$$

Case 2 ($M - j = 1$). Consider

$$\begin{aligned} M - j = 1 &\Longrightarrow \\ \mathscr{D} = \mathscr{D}^s &\cup B_M. \end{aligned} \tag{2}$$

Require:
The uncertain dataset $\mathscr{D}$ including exclusiveness rules;
The parameter k;
Ensure:
The reduced uncertain dataset $\mathscr{D}^s$;
(1) Initialize: $T' = \phi$, $T^r = \phi$, $\mathscr{D}^s = \phi$, $t_{\text{lb}}^{\max}$ = null;
(2) Re-organize $\mathscr{D}$: $\mathscr{D} \leftarrow \{B_1, B_2, \ldots, B_M\}$; //exclusiveness rules
(3) for (each B_i in $\mathscr{D}$)
(4) $\quad t_{B_i}^{\max} \leftrightarrow B_i$;
(5) $\quad T' \leftarrow T' \cup t_{B_i}^{\max}$;
(6) end for //T' is obtained
(7) Sort T';
(8) $T^r \leftarrow \{t_{B_1}^{\max}, t_{B_2}^{\max}, \ldots, t_{B_k}^{\max}\}$; //the original T^r
(9) $t_{\text{lb}}^{\max} \leftarrow t_{k+1}^{\max}$ // $\Leftrightarrow$ lb $\leftarrow k+1$
(10) for ($i = 0$; $i < k$; $i++$)
(11) $\quad$ if ($\Pr(t_{B_i}^{\max}) = 1$) then
(12) $\quad\quad$ continue;
(13) $\quad$ else
(14) $\quad\quad$ if ($\min(\mathscr{F}_{t \in B_i}(t)) > \mathscr{F}(t_{\text{lb}})$) then
(15) $\quad\quad\quad$ continue;
(16) $\quad\quad$ else
(17) $\quad\quad\quad$ while (lb $\leq M$)
(18) $\quad\quad\quad\quad T^r \leftarrow T^r \cup t_{\text{lb}}^{\max}$;
(19) $\quad\quad\quad\quad$ if ($\Pr(t_{\text{lb}}^{\max}) = 1$) then
(20) $\quad\quad\quad\quad\quad$ break;
(21) $\quad\quad\quad\quad$ else
(22) $\quad\quad\quad\quad\quad$ if (($\min(\mathscr{F}_{t \in B_{\text{lb}}}(t)) > \mathscr{F}(t_{B_{\text{lb}+1}}^{\max})$) $\|$ ($\min(\mathscr{F}_{t \in B_i}(t)) > \mathscr{F}(t_{B_{\text{lb}+1}}^{\max})$))
(23) $\quad\quad\quad\quad\quad\quad$ break;
(24) $\quad\quad\quad\quad\quad$ else
(25) $\quad\quad\quad\quad\quad\quad$ lb $\leftarrow$ lb + 1;
(26) $\quad\quad\quad\quad\quad$ end if
(27) $\quad\quad\quad\quad$ end if
(28) $\quad\quad\quad$ end while
(29) $\quad\quad$ end if
(30) $\quad$ end if
(31) end for //the extended T^r.
(32) for ($t_{B_i}^{\max} \in T^r$)
(33) $\quad \mathscr{D}^s \leftarrow \mathscr{D}^s \cup B_i$; //using $t_{B_i}^{\max} \leftrightarrow B_i$
(34) end for //$\mathscr{D}^s$ is obtained.
(35) **return** $\mathscr{D}^s$;

Algorithm 1: The MSS4UTop-k algorithm.

Suppose $B_M = \{t_{B_M}^1, t_{B_M}^2, \ldots, t_{B_M}^l\}$; then

$$\mathscr{PW} = \left\{ \mathscr{PW}_{\mathscr{D}^s}^i \cup t_{B_M}^j \mid 1 \le i \le \frac{m}{l},\ 1 \le j \le l \right\}. \quad (3)$$

For any $t_{B_M}^j$, it has no chance to enter the top-k result set, so we can conclude that Top-$k(\mathscr{PW})$ = Top-$k(\mathscr{PW}_{D^s})$; that is,

$$\mathscr{T} = \mathscr{T}_{D^s}. \quad (4)$$

Assume that $T_{D^s}^i \in \mathscr{T}_{D^s}$, according to (4), $\exists T^j \in \mathscr{T}$, and $T_{D^s}^i = T^j$. Since the tuples from B_M have no influence on the top-k record set in any of the possible worlds, $\text{PW}(T^j) = \{\text{PW}_{D^s}(T_{D^s}^i) \cup t_{B_m}^x \mid 1 \le x \le l\}$. So,

$$\begin{aligned} &\Pr\left(\text{PW}\left(T^j\right)\right) \\ &\quad = \Pr\left(\left\{\text{PW}_{D^s}\left(T_{D^s}^i\right) \cup t_{B_m}^x \mid 1 \le x \le l\right\}\right). \end{aligned} \quad (5)$$

In addition, $\sum_{x=1}^{l} \Pr(t_{B_m}^x) = 1$. Then, we have

$$\Pr\left(T_{D^s}^i\right) = \Pr\left(T^j\right). \quad (6)$$

Finally, with the formula (4) and (6), we can conclude

$$T^* = T_{\mathscr{D}^s}^*. \quad (7)$$

Require:
The uncertain dataset $\mathscr{D}$ including exclusiveness rules;
The parameter k;
Ensure:
The reduced uncertain dataset $\mathscr{D}^s$;
(1) Initialize: $T' = \phi$, $T^r = \phi$, $\mathscr{D}^s = \phi$, $t_{\text{lb}}^{\max}$ = null;
(2) Re-organize $\mathscr{D}$: $\mathscr{D} \leftarrow \{B_1, B_2, \ldots, B_M\}$; //exclusiveness rules.
(3) for (each B_i in $\mathscr{D}$)
(4) $\quad t_{B_i}^{\max} \leftrightarrow B_i$;
(5) $\quad T' \leftarrow T' \cup t_{B_i}^{\max}$;
(6) end for //T' is obtained
(7) Sort T';
(8) $i \leftarrow 1$;
(9) $j \leftarrow 0$;
(10) while ($j \le k$)
(11) $\quad$ if ($\Pr(t_{B_i}^{\max}) = 1$) then
(12) $\quad\quad j \leftarrow j + 1$;
(13) $\quad$ end if
(14) $\quad T^r \leftarrow T^r \cup \{t_{B_i}^{\max}\}$;
(15) $\quad i \leftarrow i + 1$
(16) end while //T^r is obtained.
(17) for ($t_{B_i}^{\max} \in T^r$)
(18) $\quad \mathscr{D}^s \leftarrow \mathscr{D}^s \cup B_i$; //using $t_{B_i}^{\max} \leftrightarrow B_i$.
(19) end for //$\mathscr{D}^s$ is obtained.
(20) **return** $\mathscr{D}^s$;

Algorithm 2: The quick MSS4UTop-k algorithm.

Case 3 ($M - j > 1$). Consider

$$\begin{aligned} M - j > 1 \Longrightarrow \\ \mathscr{D} &= \mathscr{D}^s \cup B_{j+1} \cup B_{j+2} \cup \cdots \cup B_M \\ &= \left(\cdots\left(\left(\mathscr{D}^s \cup B_{j+1}\right) \cup B_{j+2}\right) \cup \cdots\right) \cup B_M. \end{aligned} \tag{8}$$

Essentially, Case 3 is the extension of Case 2, and it can be proved by repeating the steps in Case 2. □

In this section, we describe our basic MSS4UTop-k algorithm in detail. Considering the different features of original uncertain dataset, the efficiency of MSS4UTop-k varies. In next section, we will introduce a variation of the MSS4UTop-k algorithm, called Quick MSS4UTop-k. The goal of Quick MSS4UTop-k is to balance the accuracy of minimum scan scope and the efficiency of MSS4UTop-k.

4.2. The Quick MSS4UTop-k Algorithm. In the basic MSS4UTop-k algorithm, we traverse T' from the element $t_{B_1}^{\max}$ to the element $t_{B_k}^{\max}$, and for each element we judge its possibility to be in Top-k record set so as to determine whether T^r should be extended. This procedure may result in a large amount of comparison operations and decrease the efficiency of MSS4UTop-k. So, we propose the Quick MSS4UTop-k algorithm to simplify the generation procedure of T^r.

Given a uncertain database $\mathscr{D}$ and a UTop-k query on $\mathscr{D}$, we denote $\mathscr{D}^d$ as the set composed of the deterministic tuples in $\mathscr{D}$; that is, $B^d = \{t_{B_i} \mid t_{B_i} \in \mathscr{D} \text{ and } \Pr(t_{B_i}) = 1\}$. The quick MSS4UTop-$k$ algorithm is suitable in the situations where (1) $|\mathscr{D}^d| > k$ and (2) the deterministic tuples are uniformly distributed in $\mathscr{D}$. The quick MSS4UTop-k algorithm is also composed of three phrases.

(i) Phrase 1 (determine T'). In this phrase, the sorted representative tuple set T' is produced. We omit it because this procedure is totally the same as that in the basic MSS4UTop-k algorithm.

(ii) Phrase 2 (determine T^r). In this phrase, the quick MSS4UTop-k algorithm is different from the basic one. In the quick MSS4UTop-k algorithm, we locate the kth deterministic tuple in T'. We denote it as $t_{B_{k'}}^{\max}$. Obviously, $k' \ge k$ and $\Pr(t_{B_{k'}}^{\max}) = 1$. Then, $T^r = \{t_{B_1}^{\max}, t_{B_2}^{\max}, \ldots, t_{B_k}^{\max}, \ldots, t_{B_{k'}}^{\max}\}$.

(iii) Phrase 3 (determine $\mathscr{D}^s$). This phrase is also the same as that of the basic MSS4UTop-k algorithm; that is, $\mathscr{D}^s = \bigcup_{t_{B_i}^{\max} \in T^r} B_i$.

Compared with that in the basic MSS4UTop-k algorithm, T^r in the quick MSS4UTop-k algorithm might be enlarged. However, if the deterministic tuples are uniformly distributed in $\mathscr{D}$ and $|\mathscr{D}^d|$ is close to $|\mathscr{D}|$, the number of redundant tuples in T^r will be rational and acceptable. Algorithm 2 describes the detailed procedure of the algorithm Quick MSS4UTop-k.

5. Experiments and Analysis

We built our framework on a 3.4-GHz Pentium IV PC with 8 GB main memory. Both the MSS4UTop-k and quick MSS4UTop-k algorithm are implemented in Java and based on the database MySQL5.5. We conducted extensive experiments for two goals: (1) to examine how data distribution and the parameter k affect the minimum scan scope for UTop-k processing; (2) to examine what the performance of the MSS4UTop-k and quick MSS4UTop-k algorithm in time consumption is.

5.1. Methodology

Data Set. Our experiment is based on the statistics that describes the driving profiles in expressway of thousands of drivers. It is derived from G15-Expressway Vehicle Speed Monitoring Database. G15 is 387 kilometers long totally, along which there are 13 vehicle speed measurement stations. Each of the measurement stations is equipped with speed detecting radar and HD cameras. When a vehicle passes by, its instant speeds as well as the license plate and timestamps will be recorded and transmitted back to the center database. The upper speed limit of G15 is 120 KM/h for small/medium-sized motor vehicles and 100 KM/h for large-sized vehicles.

Traffic insurance companies use the statistical data derived from the Expressway Vehicle Speed Monitoring Database to illustrate the driving profile of drivers. They group speed values of a car according to three predefined ranges: (1) ≥130 for small/medium-sized motors and ≥110 for large-sized vehicles. Speeds in this area mean an extremely dangerous driving behavior: (2) (120, 130) for small/medium-sized motors and (100, 110) for large-sized motors. Speeds in this area mean a dangerous driving behavior: (3) ≤120 for small/medium-sized motors and ≤100 for large motors. Speeds in this area mean a normal-driving behavior. So, given thirteen records of a car, the insurance company will calculate the average speed and frequency at each range; then they obtain a driver's driving profile. For example, Table 2 is the speed records of a car on G15. According to Table 2, the driver's profile is $\{t_B^{\max} = \langle 134.1, 1/13\rangle, \langle 123.8, 2/13\rangle, t_B^{\min} = \langle 101.5, 10/13\rangle\}$.

We use the records on September 22, 2010, in our experiment. We choose the day of September 22, 2010, because the weather was nice and traffic was flowing smoothly on that day. Studies indicate that under these two conditions a driver's driving habit will not be influenced by other drivers. In addition, on G15 the shortest distance between two neighbored monitoring stations is 2.5 KM, which means that the thirteen speed records in dataset for each vehicle can be regarded as being independent. Actually, there were totally 53,717 vehicles that ever ran on G15 on September 22, 2010. We picked out 4823 vehicles of them according to the following two principles: (1) Each vehicle must finish a complete single trip on G15 and (2) its thirteen licence plate photos must be clear enough to be accurately recognized. Next, We use the speed records of these 4823 vehicles to generate the statistical data of driver's driving profile, as depicted in Table 3.

Table 2: An example of speed records.

RadarID	Speed (KM/h)
G15N001	82.1
G15N002	111.5
G15N003	124.9
G15N004	114.1
G15N005	134.1
G15N006	96.2
G15N007	122.7
G15N008	100.0
G15N009	109.3
G15N010	119.3
G15N011	98.0
G15N012	104.5
G15N013	80.7

Table 3: Description of experiment data.

Subset	Type	Deterministic?	Number
$(S/M)^D$	Small/medium	Yes	1606
$(S/M)^U$	Small/medium	No	1872
L^D	Large	Yes	570
L^U	Large	No	775

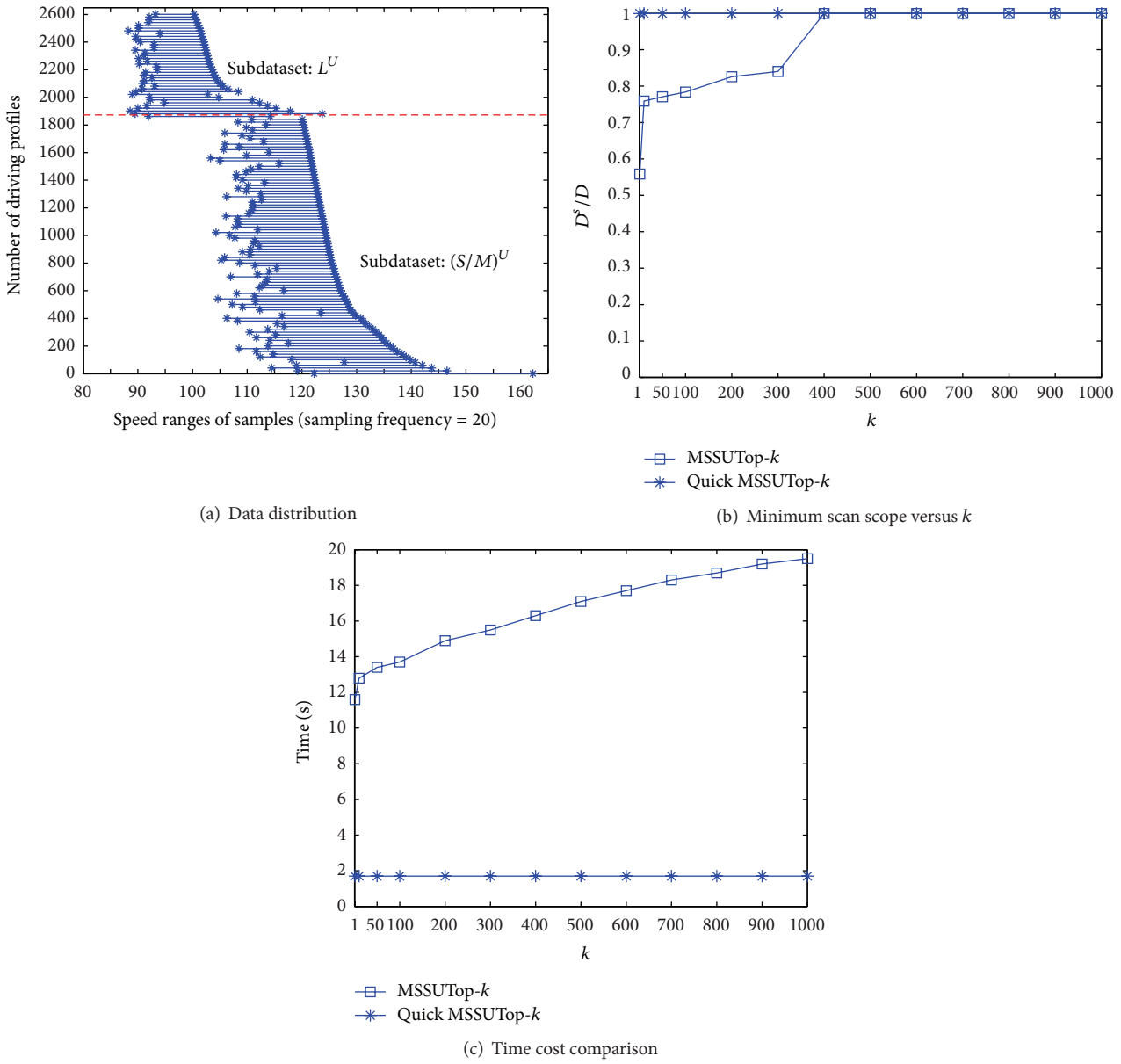
The subdataset $(S/M)^U$ and L^U is uncertain, and each vehicle may contain 2 or 3 mutually exclusive tuples. The subdatasets $(S/M)^D$ and L^D are deterministic dataset.

Methods and Evaluation Metrics. From the detailed description in Section 4, we can see that data distribution and the value of k are the two key metrics for determining the minimum scan scope for UTop-k queries. So, we test our proposed algorithm MSS4UTop-k and quick MSS4UTop-k in this paper on various datasets with different data distributions: (1) pure uncertain dataset, which is totally composed of uncertain data, and (2) mixed uncertain dataset, which is composed of uncertain data and deterministic tuples. In each of our experiments, we describe the data distribution first. Then, we measure the ratio of the minimum scan scope to the whole dataset; that is, Ratio $= \mathscr{D}^s/\mathscr{D}$. The value of k varies from 1 to 1000 uniformly. We also test the time cost of MSSUTop-k and quick MSSUTop-k respectively.

5.2. Experiment Results

5.2.1. Pure Uncertain Dataset: $(S/M)^U \cup L^U$. We combine the subdatasets $(S/M)^U$ and $(L)^U$ to compose a pure uncertain dataset, which contains the driving profiles of 2647 drivers; that is, the number of X-tuple buckets is 2647. Figure 4(a) is the data distribution of $(S/M)^U \cup L^U$. We sampled every 20 X-tuple buckets and illustrate the speed ranges of these samples. Figure 4(a) shows that data ranges overlap much.

Result Analysis. Theoretically, the scenario in Figure 4(a) is the worst case because a large amount of tuples will be involved in Top-k query processing, even if the value of k



(a) Data distribution

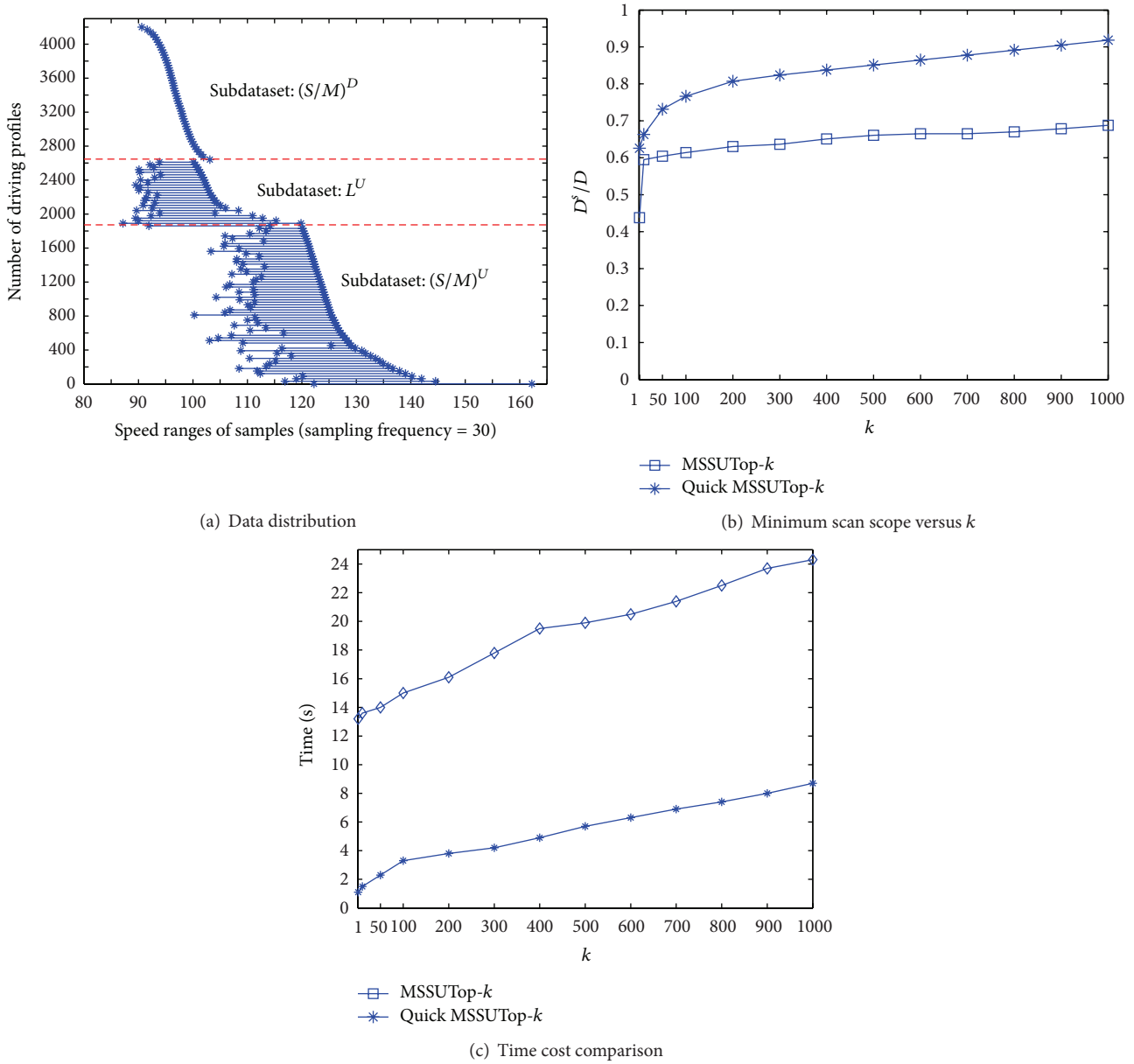
(b) Minimum scan scope versus k

(c) Time cost comparison

FIGURE 4: Data distribution, minimum scan scope, and time cost of MSSUTop-k and quick MSSUTop-k on pure uncertain dataset.

is very small. This is proved in Figure 4(b). For example, when $k = 1$, about 63% of 2647 vehicles have chance to enter the answer set. In this scenario the Extension Stop Principle 1 is invalid. Moreover, the Extension Stop Principles 2 and 3 are hard to be satisfied because the data ranges overlap much. However, we also find that D^s does not go up sharply with k when $k \geq 100$. This is because when $k \geq 100$, more large-sized vehicles are involved in the calculation of T^r. So, the Extension Stop Principles 2 and 3 are relatively easier to be satisfied. It reflects the truth that, compared with the large-sized vehicles, small/medium-sized vehicles are more inclined to be present in the final Top-k fast car set. In this scenario, quick MSSUTop-k is invalid because the number of deterministic tuples is smaller than k. So, for the pure uncertain dataset as in Figure 4(a) the scan scope of Quick MSSUTop-k algorithm is always the complete uncertain dataset. Figure 4(c) shows the time cost of MSSUTop-k and quick MSSUTop-k. For quick MSSUTop-k, it is just the time to scan the whole uncertain dataset no matter what the parameter k is. For MSSUTop-k, with the increase of k, there are more elements in original T^r. It means that the MSSUTop-k algorithm has to do more comparison operation to determine when the procedure of extension should be started and stopped. So, we can see in Figure 4(c) that the time cost of MSSUTop-k increases with k and the time cost of quick MSSUTop-k remains unchanged.

5.2.2. Mixed Dataset 1: $(S/M)^D \cup (S/M)^U \cup L^U$. We conducted our second experiment on the mixed uncertain dataset $\{(S/M)^D \cup (S/M)^U \cup L^U\}$. There are totally 4253 drivers'



(a) Data distribution

(b) Minimum scan scope versus k

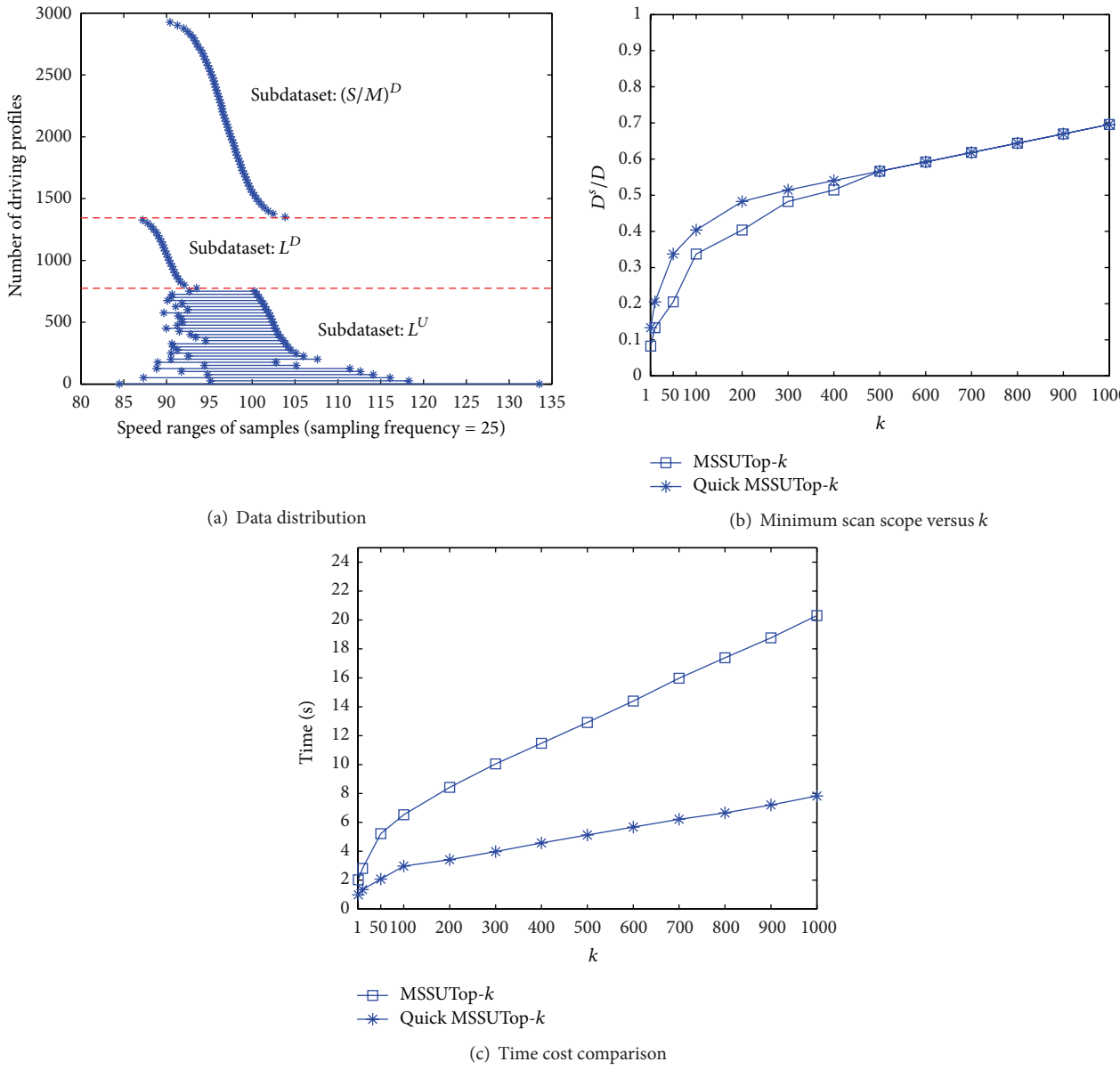
(c) Time cost comparison

FIGURE 5: Data distribution, minimum scan scope, and time cost of MSSUTop-k and quick MSSUTop-k on mixed uncertain dataset 1.

driving profiles. Figure 5(a) is the data range distribution of $(S/M)^D \cup (S/M)^U \cup L^U$. The feature of this mixed uncertain dataset is that deterministic tuples are not in uniform distribution in the whole dataset. To be specific, most values of the subdataset $(S/M)^D$ are lower than the max speed of the tuples in $(S/M)^U$. It reflects the truth that the drivers who observed the upper speed limit drive slower than the drivers who broke the upper speed limit.

Result Analysis. When k is small, this scenario is the same as scenario 1; that is, D^s increase very fast because of the densely distributed uncertain data. However, when $k \geq 50$, the deterministic tuples from $(S/M)^D$ are extended into T^r. So, the T^r will not increase sharply because the Extension Stop Principle 1 is more likely to be satisfied. Figure 5(c) is comparison of time consumption for MSSUTop-k and quick MSSUTop-k. Essentially, the quick MSSUTop-k abandons the Extension Stop Principles 2 and 3 so as to reduce the cost in comparison operation. Thus, from Figures 5(b) and 5(c), we can see that D^s of quick MSSUTop-k increases faster than that of MSSUTop-k, but quick MSSUTop-k performs better in time cost.

5.2.3. Mixed Dataset 2: $(S/M)^D \cup L^U \cup L^D$. In our third experiment we employ another mixed dataset $(S/M)^D \cup L^U \cup L^D$, which is illustrated in Figure 6(a). In this uncertain dataset, the data range of deterministic subdataset L^D is lower than that of L^U, which is similar to the Mixed Dataset 1. However, considering the subdatasets L^U and $(S/M)^D$, it is quite



(a) Data distribution

(b) Minimum scan scope versus k

(c) Time cost comparison

Figure 6: Data distribution, minimum scan scope, and time cost of MSSUTop-k and quick MSSUTop-k on mixed uncertain dataset 2.

different from the Mixed Dataset 1 in data distribution: the data range of the deterministic subdataset $(S/M)^D$ partially overlaps with that of the uncertain subdataset L^U. It means that the deterministic tuples disseminates in the uncertain tuples. It reflects the truth that the speed of furious-driving large-sized vehicles are almost the same as that of normal-driving small/medium-sized vehicles.

Result Analysis. Theoretically, there will be fewer tuples to be involved in UTop-k query processing in this case. The reason is when the deterministic tuples disseminates in the uncertain tuples, the Extension Stop Principle 1 is more inclined to be satisfied. It is proved by Figure 6(b). Initially, when k is small, the Extension Stop Principles 1, 2, and 3 are all valid. So, the ratio of minimum scan scope to the whole dataset is quite smaller than that of the other two cases above. When $k > 400$, all the uncertain tuples are involved in calculation of minimum scan scope. Hereafter, only the Extension Stop Principle 1 is valid, and the remaining tuples in T' waiting to be extended into T^r are all deterministic. So, the curves of MSSUTop-k and quick MSSUTop-k in Figure 6(b) overlap when $k > 400$. However, in time consumption, as depicted in Figure 6(c), MSSUTop-k is always more costly than quick MSSUTop-k. This is because the algorithm MSSUTop-k needs more comparison. It reminds us that, given a dataset with a small proportion of uncertain tuples, the approximate result achieved by quick MSSUTop-k may be very closer to or even the same as that of the accurate algorithm, while the time cost of quick MSSUTop-k is much less.

5.2.4. Other Scenarios. Actually, there are $15(2^4 - 1)$ combinations with the four subdatasets except for empty one. However, we just choose three typical combinations in our experiments because they can represent other situations of data distributions. For an example, the dataset $L^U \cup L^D$ has the same feature in data distribution as $(S/M)^D \cup (S/M)^U$, which is included in our second experiment. We also omit the scenarios like $(S/M)^D$, L^D, and $(S/M)^D \cup L^D$. Obviously, these three datasets are completely composed of deterministic tuples. Top-k queries on deterministic dataset have been much studied and are not the emphasis in this paper.

6. Conclusions

In this paper, we introduce two novel algorithms MSSUTop-k and quick MSSUTop-k for determining the minimum scan scope of UTop-k queries in uncertain databases. We test the performance of the proposed algorithms through extensive experiments based on real dataset. In addition, by analyzing the relationship between the minimum scan scope for processing UTop-k queries and the data distribution of various of uncertain dataset, we know that the ratio of minimum scan scope to the whole uncertain dataset varies dramatically because of different data distribution. It demonstrates that this work should be the indispensable prerequisite for UTop-k processing. By the work in this paper, given a uncertain dataset and k, users can determine exactly in advance how many tuples and which tuples will be involved for processing UTop-k queries. Then, they can make a balance between the result precision and processing cost and then choose a proper optimized solution according to the computing resources they have.

Notations

$\mathscr{F}$: Scoring function
$\mathscr{D}$: Uncertain dataset
N: Cardinality of $\mathscr{D}$
$\mathscr{PW}$: Possible worlds space
n: Cardinality of $\mathscr{PW}$
$\mathscr{T}$: Candidate UTop-k answer set
m: Cardinality of $\mathscr{T}$
$\mathrm{PW}(T^i)$: Possible worlds with T^i as top-k answer
B_i: $B_i \subseteq \mathscr{D}$, the ith X-tuple bucket
M: The number of B_is in $\mathscr{D}$
$t_{B_i}^{\max}$: The tuple in B_i with $\max(\mathscr{F}_{t_i \in B_i}(t_i))$
D^d: The subset of deterministic tuples in $\mathscr{D}$
T': The sorted X-tuple buckets
T^r: Representative tuple set
D^s: Minimum scan scope for UTop-k queries.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grant no. 61173027 and the Fundamental Research Funds for the Central Universities (N140404006).

References

[1] M. A. Soliman, I. F. Ilyas, and K. C.-C. Chang, "Top-k query processing in uncertain databases," in *Proceedings of the 23rd International Conference on Data Engineering (ICDE '07)*, pp. 896–905, IEEE, Istanbul, Turkey, April 2007.

[2] O. Benjelloun, A. D. Sarma, A. Halevy, and J. Widom, "Uldbs: databases with uncertainty and lineage," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06)*, VLDB Endowment, pp. 953–964, Seoul, Republic of Korea, September 2006.

[3] B. Babcock and C. Olston, "Distributed Top-K Monitoring," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 28–39, ACM, June 2003.

[4] I. F. Ilyas, G. Beskales, and M. A. Soliman, "A survey of top-k query processing techniques in relational database systems," *ACM Computing Surveys*, vol. 40, no. 4, article 11, Article ID 1391730, 2008.

[5] C. Li, K. C.-C. Chang, I. F. Ilyas, and S. Song, "RankSQL: query algebra and optimization for relational top-k queries," in *PRoceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '05)*, pp. 131–142, ACM, Baltimore, Md, USA, June 2005.

[6] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, "Robust and efficient fuzzy match for online data cleaning," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD/PODS '03)*, vol. 9, pp. 313–324, ACM, San Diego, Calif, USA, June 2003.

[7] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, p. 48, IEEE, April 2006.

[8] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB '04)*, vol. 30 of *VLDB Endowment*, pp. 588–599, 2004.

[9] A. Halevy, A. Rajaraman, and J. Ordille, "Data integration: the teenage years," in *Proceedings of the 32nd International Conference on Very Large Data Bases*, pp. 9–16, VLDB Endowment, Seoul, Republic of Korea, 2006.

[10] M. Magnani and D. Montesi, "A survey on uncertainty management in data integration," *Journal of Data and Information Quality*, vol. 2, no. 1, article 5, 2010.

[11] C. Re and D. Suciu, "Management of data with uncertainties," in *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM '07)*, pp. 3–8, ACM, November 2007.

[12] M. Hua, J. Pei, W. Zhang, and X. Lin, "Efficiently answering probabilistic threshold top-k queries on uncertain data," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 1403–1405, IEEE, Cancún, Mexico, April 2008.

[13] C. Jin, K. Yi, L. Chen, J. Yu, and X. Lin, "Sliding-window top-k queries on uncertain streams," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 301–312, 2008.

[14] F. Li, K. Yi, and J. Jestes, "Ranking distributed probabilistic data," in *Proceedings of the International Conference on Management of Data (SIGMOD-PODS '09)*, pp. 361–374, July 2009.

[15] G. Cormode, F. Li, and Y. Ke, "Semantics of ranking queries for probabilistic data and expected ranks," in *Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE '09)*, pp. 305–316, IEEE, Shanghai, China, April 2009.

[16] M. Dallachiesa, T. Palpanas, and I. F. Ilyas, "Top-k nearest neighbor search in uncertain data series," *Proceedings of the VLDB Endowment*, vol. 8, no. 1, pp. 13–24, 2014.

[17] M. Dylla, I. Miliaraki, and M. Theobald, "Top-k query processing in probabilistic databases with non-materialized views," in *Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE '13)*, pp. 122–133, IEEE, Brisbane, Australia, April 2013.

[18] H. T. Nguyen and J. Cao, "Trustworthy answers for top-k queries on uncertain big data in decision making," *Information Sciences*, vol. 318, pp. 73–90, 2015.

[19] T. Chen, L. Chen, M. T. Özsu, and N. Xiao, "Optimizing multi-top-k queries over uncertain data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 8, pp. 1814–1829, 2013.

[20] D. Liu, C. Wan, N. Xiong, J. H. Park, and S. Rho, "Top-*k* entities query processing on uncertainly fused multi-sensory data," *Personal and Ubiquitous Computing*, vol. 17, no. 5, pp. 951–963, 2013.

[21] K. Yi, F. Li, G. Kollios, and D. Srivastava, "Efficient processing of top-*k* queries in uncertain databases with x-relations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 12, pp. 1669–1682, 2008.

[22] M. A. Soliman, I. F. Ilyas, D. Martinenghi, and M. Tagliasacchi, "Ranking with uncertain scoring functions: semantics and sensitivity measures," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '11)*, pp. 805–816, ACM, Athens, Greece, June 2011.

[23] L. Mo, R. Cheng, X. Li, D. W. Cheung, and X. S. Yang, "Cleaning uncertain data for top-*k* queries," in *Proceedings of the 29th International Conference on Data Engineering (ICDE '13)*, pp. 134–145, IEEE, Brisbane, Australia, April 2013.

[24] J. Xu, D. V. Kalashnikov, and S. Mehrotra, "Efficient summarization framework for multi-attribute uncertain data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '14)*, pp. 421–432, ACM, Snowbird, Utah, USA, June 2014.