Identifying Top k Dominating Objects over Uncertain Data

Liming Zhan, Ying Zhang, Wenjie Zhang, and Xuemin Lin

University of New South Wales, Sydney NSW, Australia
{zhanl,yingz,zhangw,lxue}@cse.unsw.edu.au

Abstract. Uncertainty is inherent in many important applications, such as data integration, environmental surveillance, location-based services (LBS), sensor monitoring and radio-frequency identification (RFID). In recent years, we have witnessed significant research efforts devoted to producing probabilistic database management systems, and many important queries are re-investigated in the context of uncertain data models. In the paper, we study the problem of top k dominating query on multi-dimensional uncertain objects, which is an essential method in the multi-criteria decision analysis when an explicit scoring function is not available. Particularly, we formally introduce the top k dominating model based on the state-of-the-art top k semantic over uncertain data. We also propose effective and efficient algorithms to identify the top k dominating objects. Novel pruning techniques are proposed by utilizing the spatial indexing and statistic information, which significantly improve the performance of the algorithms in terms of CPU and I/O costs. Comprehensive experiments on real and synthetic datasets demonstrate the effectiveness and efficiency of our techniques.

1 Introduction

Ranking query is an essential analytic method which focuses on retrieving the top k most important answers from massive quantity of data according to an user's preference. In many applications, users need to make decision against multiple features of the objects. For instance, cost, comfort, safety, and fuel economy may be some of the main criteria we consider when purchasing a car. Therefore, each object can be described by a point in a multi-dimensional space where each dimension corresponds to a particular selection feature. If there is a scoring (utility) function (e.g., additive linear function) which can quantify the preference of a user, objects can be immediately ranked based on their corresponding scores. However, in many scenarios users cannot find a proper scoring function due to various reasons such as the lack of domain knowledge. By utilizing the dominance relationship, the **top k dominating query** provides a simple and intuitive way to rank objects when an explicit scoring function is not available. The dominance relationship has been widely used in the multi-criteria decision analysis where an object A *dominates* another object B if A is not worse than B on every dimension and A is strictly better than B on at least one dimension. The goodness of an object A, namely *dominance score*, can be naturally

S.S. Bhowmick et al. (Eds.): DASFAA 2014, Part I, LNCS 8421, pp. 388–405, 2014.

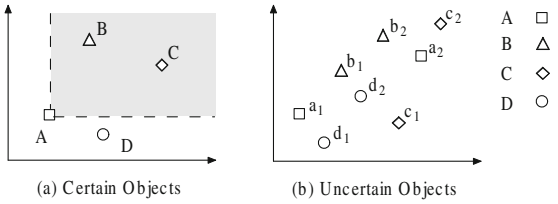


Fig. 1. Certain Objects and Uncertain Objects

measured by the number of other objects dominated by A in a set of objects, and the top k objects with highest *dominance scores* are returned as the top k dominating objects.

Example 1. As shown in Figure 1(a), there is a set $\mathcal{O}$ of 4 objects. The *dominance score* of the object A is 2 which is the number of other objects within the shaded region. Similarly, the *dominance scores* of B, C and D are 0, 0 and 1 respectively. Therefore, the results of the top 2 dominating query on $\mathcal{O}$ are A and D.

Motivation. In many applications such as data integration, environmental surveillance, location based service, the uncertainty is inherent due to many factors including limitation of measuring equipments, probabilistic model based data integration, noise, delay or loss of data updates, etc. Consequently, the data in the above applications are usually described by probabilistic model (i.e., uncertain objects) instead of deterministic points in a multi-dimensional space where an uncertain object may be described by probabilistic density function or a set of instances (points). For example, in the meteorology system, sensors collect the temperature and relative humidity at a large number of sites. The readings may be uncertain due to the noise or the limit of the sensor reader. Another example is the performance evaluation of NBA players. A player attends multiple games and statistics (e.g. score and #rebounds) of each game are recorded. Consequently, the performance of a player may be naturally modeled as an uncertain object where each record corresponds to an instance. In these applications, the top k dominating query plays an important role in multi-criteria decision analysis when the scoring function is not available. For instance, users may identify the top k risky observation sites in the meteorology system, or find the top k valuable players based on their game-by-game performance. Due to the inherent differences between uncertain data and traditional data, many important queries are re-investigated in the context of uncertain data models. In the paper, we study the problem of the top k dominating query on uncertain data.

Challenges. The main challenges of the paper are two-fold. Firstly, we need to develop a model to properly identify the top k dominating objects. Secondly, efficient computation algorithm is required to support the new top k dominating query on uncertain data.

As shown in Figure 1(b), each uncertain object may consist of multiple instances (points) , and each instance will appear with a particular probability.

For example, the uncertain object A consists of two instances a_1 and a_2. Due to the existence of multiple instances of the uncertain objects, it is non-trivial to measure the number of other uncertain objects dominated by an uncertain object (i.e., *dominance score*) since the traditional dominance relationship is defined against two points. Intuitively, we can derive the *dominance score* of an instance u based on the probability mass of the instances which are dominated by u. Then the problem of the top k dominating query can be mapped to the problem of the top k query on uncertain data since each multi-dimensional uncertain object corresponds to a score distribution based on the *dominance score* and appearance probabilities of their instances. The problem of the top k query on uncertain data has been extensively studied in recent years, and many models are proposed. Particularly, as shown in [9] the *parameterized ranking* function can unify most popular ranking semantics, and hence it is widely used as the de facto top k semantics for uncertain data. Although there are some existing work [11,10,20] investigating the top k dominating query on uncertain data, none of them supports the *parameterized ranking* semantics. Moreover, as shown in [3] the top k semantics adopted in [11,10,20] cannot properly capture the ranking of both probabilities and values.

In the paper, we adopt the *parameterized ranking* semantics to formally define the top k dominating query on multi-dimensional uncertain objects and propose an effective and efficient algorithm to support the top k dominating query by developing novel pruning techniques based on popular R-tree based indexing structure and some simple statistics information.

Contributions. Our principal contributions in this paper can be summarized as follows.

- We formally introduce a top k dominating query for multi-dimensional uncertain objects based on the state-of-the-art top k semantics on uncertain data.
- We propose effective and efficient top k dominating computation algorithms on multi-dimensional uncertain objects based on novel pruning techniques.
- We further improve the performance of the algorithm by utilizing statistics information of the uncertain objects.
- Comprehensive experiments on real and synthetic datasets demonstrate the efficiency and scalability of our techniques.

Roadmap. The rest of the paper is organized as follows. Section 2 introduces the problem and some preliminary knowledge. Section 3 develops efficient algorithms to support the top k dominating query by utilizing the spatial indexing and statistics information. The experimental results are reported in Section 4. This is followed by the related work presented in Section 5. We conclude our paper in Section 6.

2 Preliminary

We present problem definition and necessary preliminaries in this section. Table 1 summarizes notations frequently used throughout the paper.

Table 1. The summary of notations

Notation	Meaning
U, V	uncertain objects
u, v	instances of the uncertain objects
$u \prec v$	u *dominates* v
p_u	occurrence probability of the instance u
$s(u)$	*dominance score* of the instance u
$\Upsilon(U)/\Upsilon(u)$	the *rank score* of an object U/instance u
U_{mbr}	minimal bounding rectangle of U
U^-_{mbr} (U^+_{mbr})	lower (upper) corner point of U_{mbr}
U_s	*dominance score* distribution of U
U^-_s (U^+_s)	lower (upper) bound of the *dominance scores* for instances in U

2.1 Problem Definition

A point (instance) p is in a d-dimensional space and the i-th dimensional coordinate value of p is denoted by $p.D_i$. Without loss of generality, we assume smaller coordinate values are preferred. For two points p and q, p *dominates* q, denoted by $p \prec q$, if $p.D_i \leq q.D_i$ for all dimension $i \in [1, d]$ and there is at least one dimension $j \in [1, d]$ with $p.D_j < q.D_j$. Meanwhile, we use $p \preceq q$ to denote that p *dominates or equals* q.

Uncertain Object Model. An uncertain object can be described either *continuously* or *discretely*. In the paper, we focus on the *discrete* case. Note that we can discretize a continuous probability density function (PDF) of an uncertain object by sampling methods. In the *discrete* case, an uncertain object U consists of a set $\{u_1, u_2, \ldots, u_m\}$ of instances (points). For $1 \leq i \leq m$, an instance u_i occurs with probability p_{u_i} ($p_{u_i} > 0$), and $\sum_{i=1}^{m} p_{u_i} = 1$. We assume that the uncertain objects are *independent* to each other. In the following paper, we use *object* to denote *multi-dimensional uncertain object* whenever there is no ambiguity. Given an object U, U_{mbr} denotes the minimal bounding rectangle which contains all of the instances of U. Let U^-_{mbr} (U^+_{mbr}) denote the lower (upper) corner of U_{mbr}, we have $U^-_{mbr} \preceq u$ and $u \preceq U^+_{mbr}$ for any instance $u \in U_{mbr}$.

Dominance Score Distribution. Based on the dominance relationship against two points (instances), we can easily measure the goodness of an instance as follows.

Definition 1 (Dominance Score). *Given a set $\mathcal{O}$ of objects, the dominance score of an instance u of the object U, denoted by $s(u)$, is*

$$s(u) = \sum\nolimits_{V \in\ \mathcal{O} \setminus U}\ \sum\nolimits_{v \in V\ \wedge\ u \prec v} p_v \tag{1}$$

where $\sum_{v \in V \wedge u \prec v} p_v$ represents the probability that the object V is dominated by the instance u.

Given a set $\mathcal{O}$ of objects, we can derive the *dominance score distribution* for each object $U \in \mathcal{O}$, denoted by U_s, where $U_s = \{(s(u_1), p_{u_1}), \ldots, (s(u_i), p_{u_i}), \ldots,$

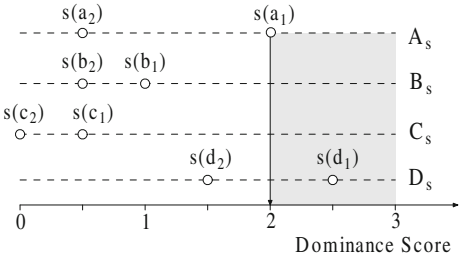


Fig. 2. Dominance Score Distributions

$(s(u_m), p_{u_m})\}$. We use $Pr(U_s > c)$ to denote the probability that U_s is larger than the value c, where $Pr(U_s > c) = \sum_{u_i \in U \wedge s(u_i) > c} p_{u_i}$.

Example 2. Regarding the example in Figure 1(b) and Definition 1, *dominance score distributions* of four objects A, B ,C and D are depicted in Figure 2 where we assume each instance has appearance probability 0.5. Particularly, as a_1 dominates four instances (b_1, b_2, d_2, and c_2), and a_2 only dominates c_2, we have $s(a_1) = 2$, $s(a_2) = 0.5$. Since $p_{a_1} = p_{a_2} = 0.5$, we get $A_s = \{(2, 0.5), (0.5, 0.5)\}$. Similarly, we have $B_s = \{(1, 0.5), (0.5, 0.5)\}$, $C_s = \{(0.5, 0.5), (0, 0.5)\}$, $D_s = \{(2.5, 0.5), (1.5, 0.5)\}$, and $Pr(D_s > 2) = p_{d_1} = 0.5$.

Parameterized Ranking. The *dominance score distributions* of a set of objects can be ranked by the top k semantics studied for uncertain data in the literature. In the paper, we focus on the *parameterized ranking* function (PRF^{ω}) proposed in [9] since it can unify other popular ranking functions.

For a set $\mathcal{O}$ of objects, a possible world W proposed in [4] is a set of instances with one instance from each uncertain object. Given a set of uncertain objects $\{U_1, U_2, \ldots, U_n\}$, a possible world $W = \{u_1, u_2, \ldots, u_n\}$ is a set of instances sequentially sampled from each object. Assume the uncertain objects are independent to each other, and the probability of W to appear is $Pr(W) = \prod_{i=1}^{n} p_{u_i}$. In each world W, an object is ranked based on the score (value) of its corresponding instance in W. In the paper, we use $r_W(U)$ to denote the rank of an object in the possible world W which is abbreviated to $r(U)$ whenever there is no ambiguity. Let $\mathcal{W}$ denote the set of all possible worlds, and we have $\sum_{W \in \mathcal{W}} Pr(W) = 1$ where $Pr(W)$ is the occurring probability of the possible world W. Then we have the formal definition of *parameterized ranking* function.

Definition 2 (PRF^{ω})**.** *Let ω be a weighted function which maps an object-rank pair to a complex number, the* ***rank score*** *of an object U, denoted by $\Upsilon(U)$, is defined as follows.*

$$\Upsilon(U) = \sum\nolimits_{i \in [1,n]} \omega(i) \times Pr(r(U) = i) \tag{2}$$

where $\omega(i)$ denotes the weight of the i-th position, and $Pr(r(U) = i)$ denotes the probability of U ranked at the i-th position, i.e., $Pr(r(U) = i) = \sum_{W \in \mathcal{W} \wedge r(U) = i} Pr(W)$. Recall that $r(U)$ denotes the rank of U in the possible world W.

In the paper, we assume $w(i) \geq w(j)$ for any two ranking positions i and j where $i < j$. This is very intuitive as a higher position is usually at least as desirable as those behind it and thus should be given a higher weight.

Problem Statement. Given a set $\mathcal{O}$ of objects, we aim to return the top k dominating objects based on their *dominance score distributions* and *parameterized ranking* semantics; that is, we retrieve the top k objects with the highest *rank scores* regarding their *dominance score distributions*.

2.2 Computing Rank Score $\Upsilon(U)$

As shown in [9], the *rank score* of an object U can be calculated by the summation of the *rank scores* of its instances. For an instance $u \in U$, we can use the following *generating function* $\mathcal{F}(\mathbf{x}, u)$ to calculate the *rank score* of u, where $P_{V,u}$ is the probability that V_s is larger than $s(u)$ (i.e., $P_{V,u} = Pr(V_s > s(u)) = \sum_{v \in V \wedge s(v) > s(u)} p_v$). Recall that V_s is the *dominance score distribution* of the object V. Intuitively, a small $P_{V,u}$ is in favor of the *rank score* of the instance u.

$$\mathcal{F}(\mathbf{x}, u) = \left(\prod\nolimits_{V \in \mathcal{O} \setminus U} (1 - P_{V,u} + P_{V,u} \cdot \mathbf{x}) \right) \ (p_u \cdot \mathbf{x}) \tag{3}$$

Then we have $Pr(r(u) = i) = c_i$ where $r(u)$ is the rank position of instance u and c_i is the coefficient of $\mathbf{x}^i$ in $\mathcal{F}(\mathbf{x}, u)$. Therefore, after applying Equation 2, we have

$$\Upsilon(u) = \sum\nolimits_{1 \leq i \leq n} \omega(i) \times c_i \tag{4}$$

$$\Upsilon(U) = \sum\nolimits_{u \in U} \Upsilon(u) \tag{5}$$

Example 3. In Figure 2, we have $s(a_1) = 2$ and hence $Pr(B_s > s(a_1)) = 0.0$, $Pr(C_s > s(a_1)) = 0.0$, and $Pr(D_s > s(a_1)) = 0.5$ (i.e., the probability mass of the instances of D in the shaded area). According to Equation 3, $\mathcal{F}(\mathbf{x}, a_1) = (1) \times (1) \times (0.5 + 0.5\ \mathbf{x}) \times 0.5\ \mathbf{x} = 0.25\ \mathbf{x} + 0.25\ \mathbf{x}^2$. Therefore, we have $Pr(s(a_1) = 1) = 0.25$ and $Pr(s(a_1) = 2) = 0.25$. Similarly, $\mathcal{F}(\ \mathbf{x}, a_2) = (0.5 + 0.5\ \mathbf{x}) \times (1) \times (\mathbf{x}) \times 0.5\ \mathbf{x} = 0.25\ \mathbf{x}^2 + 0.25\ \mathbf{x}^3$, and hence $Pr(s(a_2) = 1) = 0$, $Pr(s(a_2) = 2) = 0.25$,and $Pr(s(a_2) = 3) = 0.25$. Suppose $\omega(1) = 4$, $\omega(2) = 3$, $\omega(3) = 2$ and $\omega(4) = 1$ in PRF^{ω}, then $\Upsilon(A) = \Upsilon(a_1) + \Upsilon(a_2) = (0.25 \times 4 + 0.25 \times 3) + (0.25 \times 3 + 0.25 \times 2) = 3$ according to Equation 4 and 5. Similarly, we have $\Upsilon(B) = 2.5$, $\Upsilon(C) = 1.5$ and $\Upsilon(D) = 3.75$. Therefore, $\Upsilon(D) > \Upsilon(A) > \Upsilon(B) > \Upsilon(C)$.

3 Approach

A straightforward solution for the problem of top k dominating query is to first calculate the *dominance score* for each instance of the objects by conducting dominance checks against the instances of other objects, then compute the *rank*

scores of the objects. The k objects with the highest *rank scores* are the top k dominating objects. However, it is cost-inhibitive because the cost of *dominance score* computation is not cheap and the total number of instances may be huge. In this section, we propose efficient algorithms to identify the top k dominating uncertain objects. Specifically, Section 3.1 presents the framework of our algorithms following the *filtering and verification* paradigm. Section 3.2 proposes efficient algorithms for the computation of the *dominance score*. Section 3.3 introduces the spatial pruning technique based on the MBRs of the objects. Rank score based pruning technique is proposed in Section 3.4. In Section 3.5, statistics based approach further improves the performance of the algorithms by utilizing the probabilistic inequalities.

To facilitate the top k dominating query, we assume uncertain objects are organized by R-tree indexing techniques. Given a set of uncertain objects, the MBRs of the objects are indexed by R-tree [7], which is called the *global* R-tree.As to each uncertain object U, an *aggregate* R-tree [14] is employed to organize the instances where the aggregate value of each intermediate entry is the probability mass of the instances in the entry, which is called a *local* R-tree for an uncertain object U.

3.1 Compute Top k Dominating Objects

In this subsection, we introduce the framework of the top k dominating algorithm based on the *filtering and verification* paradigm in Algorithm 1. Pruning techniques are developed to reduce computational cost by eliminating non-promising objects. Assume MBRs of the objects are organized by the *global* R-tree $\mathcal{R}$, Line 1 derives the lower and upper bounds of the *dominance scores* for each object U, denoted by U_s^- and U_s^+ respectively (See Section 3.2). Then Line 2 applies the spatial based pruning technique to identify the candidate objects which are kept in a set $\mathcal{C}$. In addition to the candidate objects, $\mathcal{L}$ keeps the objects whose instances may contribute to the *rank scores* computation of candidate objects. As shown in Section 3.3, we can safely remove remaining objects (i.e., $\mathcal{O} \setminus \mathcal{L}$) from *rank scores* computation [1]. A max-heap $\mathcal{H}$ is employed to guarantee that instances are accessed by decreasing order, which is initialized by objects in $\mathcal{L}$ where the upper bounds of their *dominance scores* are key values in the heap(Line 4). The *generating function* $\mathcal{F}(x, u)$ is maintained to compute the *rank scores* for instances accessed (Line 8 and 10). Algorithm 1 maintains a *rank score* threshold λ to prune non-promising objects in Line 11 and 17 by utilizing *rank score* based pruning techniques (See details in Section 3.4). Line 18 loads the instances of an object and calculate their *dominance scores*, where an efficient *dominance scores* computation algorithm is presented in Section 3.2. Then Line 19 pushes these instances into the heap $\mathcal{H}$ for further processing. Line 5 terminates the algorithm when there is no candidate object for further exploration or the heap is empty. Finally the k objects with the highest *rank scores* are returned as the top k dominating objects.

[1] These objects may be involved in the *dominance score* computation of other objects.

Algorithm 1. Top k Dominating Objects ($\mathcal{R}$, k)

Input : $\mathcal{R}$: the *global* R-tree of $\mathcal{O}$, k: objects retrieved
Output : Top k dominating objects

```
1  λ := 0; Compute U_s^+ and U_s^- for all objects U ∈ O ;
2  L, C ← spatial based pruning against O ;
3  for each U ∈ L do
4      Push U into H with key value U_s^+ ;
5  while H ≠ ∅ or C ≠ ∅ do
6      E ← deheap(H);
7      if E is an instance u from object U then
8          Update the generating function F(x, u) ;
9          if U ∈ C then
10             Compute Υ(u) based on F(x, u) ;
11             C := C \ U if U can be pruned by λ ;
12             if u is the last instance of U then
13                 Compute Υ(U); Update λ; C := C \ U ;
14     else
15         E corresponds to the object U ;
16         if U can be pruned based on λ then
17             C := C \ U ;
18         Compute dominance scores for instances of U ;
19         Push every u ∈ U into H with key value s(u) ;
20 return Top k objects with highest rank scores
```

3.2 Compute Dominance Scores

In this subsection, we introduce efficient *dominance scores* computation algorithms based on the R-tree structure.

Dominance Relationships for Rectangles. A pair of uncertain objects may have three relationships as follows. Let R^+ and R^- denote the upper and lower corners of a rectangle R, we say the rectangle R_1 *fully dominates* another rectangle R_2 if $R_1^+ \prec R_2^-$. Similarly, we have R_1 *partially dominates* R_2 if $R_1^- \prec R_2^+$ and $R_1^+ \not\prec R_2^-$. Otherwise, we say R_1 does not dominate R_2. It is immediate that we have $x \prec y$ for any points $x \in R_1$ and $y \in R_2$ if R_1 *fully dominates* R_2. Similarly, $x \not\prec y$ for any points $x \in R_1$ and $y \in R_2$ if R_1 does not *dominate* R_2.

Compute the Lower and Upper Bounds (U_s^-, U_s^+)
Based on the definition of the *dominance score* (Definition 1) and the *dominance relationship* between two rectangles, we can derive the lower and upper bounds of the *dominance scores* of the instances from an object U, denoted by U_s^+ and U_s^- respectively, based on the MBRs of the objects. In this subsection, U_s^- equals the number of other objects whose MBRs are *fully dominated* by U_{mbr}. Similarly, U_s^+ is the number of other objects whose MBRs are *fully dominated* or *partially dominated* by U_{mbr}.

Motivation. We may issue two dominating range queries based on the lower and upper corners of the MBR of each object against the *global* R-tree $\mathcal{R}$ in Algorithm 1 to compute their lower and upper bounds of the *dominance scores*. Nevertheless, we can improve the computational cost by conducting a spatial join based computation. We conduct the dominance checks in a level-by-level fashion such that the dominance count can be calculated at higher level, and hence significantly reduce the number of dominance checks.

Algorithm. Algorithm 2 illustrates the details of the *dominance score* bounds computation based on MBRs of the objects, which follows the *synchronized* R-tree traversal paradigm used in spatial join. For each entry E (data entry or intermediate entry) of the *global* R-tree $\mathcal{R}$, we use a tuple T to record its lower and upper bounds of its *dominance score*, denoted by T_s^- and T_s^+ respectively, while $T.owner$ refers to the entry E. Clearly, we do not need to further explore another entry E_1 regarding E if E *fully dominates* E_1 or E does not *dominate* E_1. We use $T.set$ to keep a set of entries which are *partially dominated* by E. A FIFO queue $\mathcal{Q}$ is employed to maintain the tuples, and $\mathcal{Q}$ is initialized by a tuple T where $T.owner$ and $T.set$ are set to the root of the *global* R-tree $\mathcal{R}$ (Line 1-2). For each tuple T popped from $\mathcal{Q}$, if the entries from $T.owner$ and $T.set$ are data entries, then we have $U_s^+ = T_s^+$ and $U_s^- = T_s^-$ where $T.owner$ refers to the MBR of the object U. Otherwise, Line 7-17 expand $T.owner$ and $T.set$ for *dominance score* computation of the lower level entries. For presentation simplicity, the child entries of a data entry are referred to the data entry itself (Line 7 and 11). Specifically, for each pair of entries $t.owner$ and e at Line 12 and 13, the lower bound t_s^- and upper bound t_s^+ are increased by $e.cn$ if t_{mbr} *fully dominates* e_{mbr}, where $e.cn$ is the aggregate number of objects in entry e (Line 15). Otherwise, we only increase t_s^+ and keep e in $t.set$ for further computation if t_{mbr} *partially dominates* e_{mbr} (Line 17). Algorithm 2 terminates when $\mathcal{Q}$ is empty, and the lower and upper *dominance score* bounds of the objects are ready for the spatial pruning in Section 3.3.

Compute the Dominance Score

In the top k dominating algorithm (Line 18 of Algorithm 1) we need to compute the *dominance scores* of the instances for the object U. We can come up with an efficient *dominance scores* computation algorithm for an object U by slightly modifying Algorithm 2. Suppose U_s^- is calculated, we only need to consider a set $\mathcal{S}$ of objects which are *partially dominated* by U. For each object $V \in \mathcal{S}$, we set $T.owner$ and $T.set$ to the roots of R_U and R_V respectively at Line 1 where R_U and R_V are the local R-tree of the objects U and V respectively. When algorithm terminates, for each instance $u \in U$ we have the probability mass of instances from V which are dominated by u, denoted by $P(u, V)$. According to Definition 1, the *dominance score* of the instance u can be calculated as follows.

$$s(u) = U_s^- + \sum\nolimits_{V \in \mathcal{S}} P(u, V). \tag{6}$$

3.3 Spatial Pruning Technique

Considering that the cost is expensive to compute *dominance scores*, we propose effective spatial pruning techniques to reduce the number of candidate objects based on the *global* R-tree; that is, we aim to prune a set of objects from *dominance score* computation without accessing the instances of the objects such that the CPU and I/O costs can be significantly reduced.

Theorem 1. *Given two objects U and V, we have $\Upsilon(U) > \Upsilon(V)$ if $U_s^- > V_s^+$.*

Proof. Since $U_s^- > V_s^+$, we have $s(u) > s(v)$ for any $u \in U$ and $v \in V$. Therefore, we have $Pr(r(U) \leq i) > Pr(r(V) \leq i)$ where $Pr(r(U) \leq i)$ denotes the probability that U is ranked not lower than the i-th position. According to Equation 2 and the monotonic property of the weight function (i.e., $\omega(i) \geq \omega(j)$ for any two positions i and j with $i < j$), we have $\Upsilon(U) > \Upsilon(V)$.

Spatial Based Pruning. Let f_c denote the k-th largest lower bound of the *dominance scores* regarding objects in $\mathcal{O}$, according to Theorem 1, we have $\mathcal{C} = \{U | U \in \mathcal{O}$ and $U_s^+ \geq f_c\}$ in Algorithm 1; that is, only the object U with $U_s^+ \geq f_c$ can become the top k candidate objects. Let f_s denote the smallest lower bounds of *dominance scores* regarding objects in $\mathcal{C}$, we have $\mathcal{L} = \{U |\, U \in \mathcal{O}$ and $U_s^+ \geq f_s\}$ in Algorithm 1. Note that, as shown in Equation 3 the *rank score* of an instance can only be affected by other instances with larger *dominance scores*.

Algorithm 2. Compute Dominance Score Bounds ($\mathcal{R}$)

```
   Input  : R: the global R-tree of O
   Output : Objects with lower and upper dominance scores bounds
 1 T.owner := root of R ; T.set := root of R ;
 2 Push T into FIFO queue Q ;
 3 while Q ≠ ∅ do
 4     T ← dequeue(Q);
 5     if T.owner is not a data entry or entries in T.set are not data entries then
 6         L := ∅; W := ∅ ;
 7         for each child entry e of T.owner do
 8             t.owner = e; t_s^- := T_s^-; t_s^+ := t_s^-;
 9             L := L ∪ t ;
10         for each entry e in T.set do
11             W := W ∪ child entries of e ;
12         for each tuple t in L do
13             for each entry e in W do
14                 if t_mbr fully dominates e_mbr then
15                     t_s^- := t_s^- + e.cn; t_s^+ := t_s^+ + e.cn ;
16                 else if t_mbr partially dominates e_mbr then
17                     t_s^+ := t_s^+ + e.cn ; t.set := t.set ∪ e ;
18             Push t into Q ;
```

3.4 Rank Score Based Pruning Technique

In this subsection, we propose effective pruning techniques based on the *rank scores* of the objects. Let λ in Algorithm 1 denote the k-th largest *rank scores* for objects accessed, clearly we can safely remove an object U from the top k candidates if we can claim the upper bound of $\Upsilon(U)$ is smaller than λ.

Rank Score Based Pruning. In Algorithm 1, we invoke the *rank score* based pruning technique at Line 11 and 17. Let u_i be the i-th visited instance of U, we can calculate $\Upsilon(U)^+$ by setting $p_{u_i} = \sum_{i \leq j \leq m} u_j$ (i.e., move probability mass of the unvisited instances of U to u_i), and prune the object U from candidate set $\mathcal{C}$ at Line 11 if $\Upsilon(U)^+ \leq \lambda$. Let u_0 denote an instance where $s(u_0) = U_s^+$ and $p_{u_0} = 1.0$,i.e., an object constructed by pushing all instances of U to the lower corner of U_{mbr}. Consequently, at Line 17 of Algorithm 1, we can calculate $\Upsilon(U)^+$ based on U_s^+ without loading instances of U and remove U from candidate set $\mathcal{C}$ if $\Upsilon(U)^+ \leq \lambda$. It is immediate that we can also remove any unvisited object $V \in \mathcal{C}$ (i.e., the object V with $V_s^+ < U_s^+$) in $\mathcal{C}$ from the top k candidates since we have $\Upsilon(V) \leq \Upsilon(V)^+$ and $\Upsilon(V)^+ \leq \Upsilon(U)^+$.

3.5 Enhance the Performance with Statistics

Assume some statistics information (*mean* and *variance*) of the objects are available, we can further enhance the performance of the top k dominating query by utilizing probabilistic inequalities. Specifically, given two objects U and V, we use $\Delta^-(V,U)$ ($\Delta^+(V,U)$) to denote the contribution of U towards the lower (upper) bound of the *dominance score* of V. In Section 3.2 we have $\Delta^-(V,U) = 0$ and $\Delta^+(V,U) = 1$ if V *partially dominates* U. This subsection shows that we can derive tighter lower and upper bounds for the *dominance scores* of the objects based on their MBRs and statistics information.

Motivation. As shown in Figure 3(a), given the MBR of an object U and a point p, we use rectangle A to denote the area of U_{mbr} on the left side of p (shaded area). Similarly, in Figure 3(b), we use rectangle B to denote the area of U_{mbr} on the bottom of p (shaded area). Moreover, the rectangle R_p (the rectangle with thick line) represents the area of U_{mbr} which is dominated by the point p. Let $P(R)$ denote the probability mass of U within the rectangle R, then we should have $0 \leq \Delta^-(V,U) \leq P(R_p)$ if p corresponds to the upper corner of V_{mbr} (i.e., V_{mbr}^+). Since $P(R_p) \geq 1-(P(A)+P(B))$ in Figure 3(b), we may have $\Delta^-(V,U) = 1-(P(A)+P(B))$ .As we cannot have exact $P(A)$ ($P(B)$) value without accessing instances of U, this subsection shows that we can derive the upper bound of $P(A)$ ($P(B)$), denoted by $P^+(A)$ ($P^+(B)$) based on statistics information. Then we can come up with tight $\Delta^-(V,U)$ and $\Delta^+(V,U)$ values without loading instances of U. Specifically, we may have $\Delta^-(V,U) = 1-(P^+(A)+P^+(B))$ (p is the upper corner of V_{mbr}) and $\Delta^+(V,U) = min(P^+(A), P^+(B))$ (p is the lower corner of V_{mbr}). With similar rationale, we may have $\Delta^+(V,U) = min(P^+(A), P^+(B))$ when p corresponds to the lower corner of V_{mbr} in Figure 3(d).

Example 4. Suppose we have $P^+(A) = 0.2$ and $P^+(B) = 0.3$ in Figure 3. We have $P(R_p) \geq 1-(P^+(A)+P^+(B)) = 0.5$. Therefore, we can set $\Delta^-(V,U) = 0.5$ in Figure 3(b). Similarly, in Figure 3(d) we have $P(R_p) \leq min(P^+(A), P^+(B))$ and hence $\Delta^+(V,U)$ can be set to 0.2.

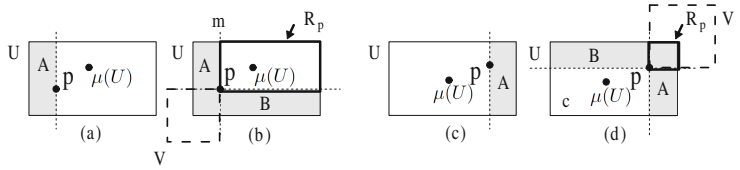


Fig. 3. Statistic based Pruning

Definitions and Lemmas. We formally define two statistics information of an object U as follows.

Definition 3 (*mean* $\mu(U)$). *We use $\mu(U)$ to denote the mean of an object U, where $\mu(U).D_i = \sum_{u \in U}(u.D_i \times p_u)$.*

Definition 4 (*variance* $\sigma^2(U)$). *$\sigma^2(U)$ denotes the variance of an object U on each dimension; that is, $\sigma_i^2(U) = \sum_{u \in U}((u.D_i - \mu(U).D_i)^2 \times p_u)$.*

Let $\delta(x,y)$ be $\frac{1}{1+\frac{x^2}{y^2}}$ if $y \neq 0$, 1 if $x = 0$ and $y = 0$, and 0 if $x \neq 0$ and $y = 0$.

Lemma 1 (Cantelli's Inequality [13]). *Suppose that t is a random variable in 1-dimensional space with mean $\mu(t)$ and variance $\sigma^2(t)$, $Prob(t-\mu(t) \geq a) \leq \delta(a, \sigma(t))$ for any $a \geq 0$, where $Prob(t - \mu(t) \geq a)$ denotes the probability of $t - \mu(t) \geq a$.*

Note that Lemma 1 extends the original Cantellis Inequality [13] to cover the case when $\sigma = 0$ and/or $a = 0$. Then we can come up with another version of Cantelli's Inequality [12], which provides an upper-bound for $Prob(t \preceq b)$ when $b < \mu$.

Lemma 2. *Assume that $0 < b < \mu(t)$. Then, $Prob(t \leq b) \leq \delta(\mu(t) - b, \sigma(t))$.*

Proof. Let $t' = 2\mu(t) - t$. It can be immediately verified that $\sigma^2(t') = \sigma^2(t)$ and $\mu(t) = \mu(t')$. The theorem holds by applying Cantelli's Inequality on t'.

Compute $\Delta^-(V,U)$. Given two objects U and V, the following theorem indicates that we can derive $\Delta^-(V,U)$ based on the statistics information.

Theorem 2. *Given two objects U and V, let p denote the upper corner of V_{mbr}, we have $\Delta^-(V,U) = 1-\sum_{1 \leq i \leq d}(\delta(\mu_i(U) - p.D_i, \sigma_i(U)))$ if $p \prec \mu(U)$. Note that we set $\delta(\mu_i(U) - p.D_i, \sigma_i(U))$ to 0 if $p.D_i < U^-_{mbr}.D_i$.*

Proof. Let R_i denote the left side of the rectangle U_{mbr} divided by the point p on the i-th dimension (e.g., $R_1 = A$ and $R_2 = B$ in Figure 3(b)). We use $P(R_i)$ to record the probabilistic mass of the instances in U contained by R_i. Clearly, we have $P(R_i) = 0$ if $p.D_i < U^-_{mbr}.D_i$ since R_i corresponds to an empty rectangle.

Otherwise, we have $P(R_i) \leq \delta(\mu_i(U) - p_i, \sigma_i(U))$ according to Theorem 2. Let R_p denote the rectangle whose lower (upper) corner is p (U^+_{mbr}), and $P(R_p)$ records the probabilistic mass of the instances in U contained by R_p. Then we have $P(R_p) \geq 1 - \sum_{1 \leq i \leq d}(\delta(\mu_i(U) - p.D_i, \sigma_i(U)))$, which implies that we can set $\Delta^-(V,U)$ to $1 - \sum_{1 \leq i \leq d}(\delta(\mu_i(U) - p.D_i, \sigma_i(U)))$ since all instances within R_p are dominated by p which is the upper corner of V_{mbr}. Therefore, the theorem holds.

Compute $\Delta^+(V,U)$. The following theorem indicates that we can derive $\Delta^+(V,U)$ based on the statistics information.

Theorem 3. *Given two objects U and V, let p denote the lower corner of V_{mbr}, we have $\Delta^+(V,U) = min(\{\delta(p.D_i - \mu_i(U), \sigma_i(U))\})$ with $1 \leq i \leq d$ if $\mu(U) \prec p$ and $p \prec U^+_{mbr}$.*

We omit the details of the proof since it is similar to Theorem 2.

Achieve Better Dominance Score bounds. Let $\hat{V}_s^-$ ($\hat{V}_s^+$) denote the new lower (upper) bound for the *dominance score* of the object V, and $\mathcal{S}$ is the set of objects which are *partially dominated* by V, we have $\hat{V}_s^- = V_s^- + \sum_{U \in \mathcal{S}} \Delta^-(V,U)$ and $\hat{V}_s^+ = V_s^- + \sum_{U \in \mathcal{S}} \Delta^+(V,U)$.

Suppose the statistics information of the objects are kept with the data entries of the objects in the *global* R-tree, we can use the new lower and upper bounds of *dominance scores* in Algorithm 1. Our empirical study shows that although the statistics information slightly increase the index size, the gain is significant since the tighter *dominance scores* bounds lead to smaller candidate size, and hence reduce the CPU and I/O costs.

4 Experiment

In this section, we present results of a comprehensive performance study to evaluate the efficiency and scalability of the proposed techniques in the paper. As there is no existing work on top k dominating query on uncertain data following the *parameterized ranking* semantics, we only evaluate the techniques proposed in the paper. Following algorithms are implemented for performance evaluation.

- **NAIVE:** The *straightforward solution* is introduced in the Section 3.
- **NAIVES:** Algorithm 1 proposed in Section 3 where only the spatial based pruning technique (Section 3.3) is employed.
- **BAS:** Algorithm 1 proposed in Section 3 where spatial based pruning technique and spatial join based *dominance score* computation algorithms (Section 3.2) are employed. It is employed as the baseline algorithm in our empirical study.
- **TKDOM: BAS** Algorithm which also applies the *rank score* based pruning technique (Section 3.4).
- **TKDOM*: TKDOM** Algorithm which also applies statistics based techniques (Section 3.5).

We employ a specific *parameterized ranking linear* function $PFR^e(\alpha)$ to rank the objects. Like the setting in [9], we use $PFR^e(\alpha = 0.95)$ in the experiments.

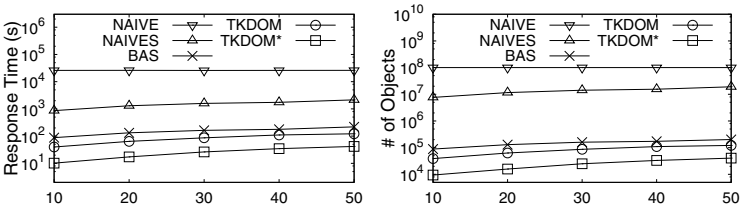
Datasets. We evaluate our techniques on both synthetic and real datasets. Synthetic datasets are generated by using the methodologies in [1] regarding the following parameters. Dimensionality d varies from 2 to 5 with default value 3. Data domain in each dimension is [0, 10000]. The number n of objects in each dataset varies from 10K to 50K with default value 10K. The number m of instances per object varies from 50 to 300 with the default value 100. The value k varies from 10 to 50 with default value 20. The edge length h of object MBRs varies from 100 to 600 with default value 200. Centers of objects follow either Equally(E), Correlated(C) or Anti-correlated(A) distribution where default is A distribution.The instances of an uncertain object follow popular distributions Normal(N) and Uniform(U) where N distribution is default. And two real datasets, Forest CoverType dataset (COV) (http://archive.ics.uci.edu/ml) and Household (HOU) (http://www.ipums.org), are employed to represent the centers of the uncertain objects. In COV, we select the horizontal and vertical distances of each observation point to the Hydrology as well as the elevation of the point. In HOU, each record represents the percentage of an American family's annual income spent on 3 types of expenditures (e.g., gas, etc.). We choose $20,000$ objects in COV and HOU respectively. For each object, we generate the instances according to the default setting above. Then with the default setting, the total number of instances in synthetic and real datasets are 1 millions and 2 millions respectively.

All algorithms are implemented in standard C++ and compiled with GNU GCC. Experiments are run on a PC with Intel Xeon 2.40GHz dual CPU and 4G memory under Debian Linux. In the paper, we evaluate the I/O performance of the algorithms by measuring the number of uncertain objects explored.

Performance Evaluation

In the first experiment, we vary the value of k and evaluate the performance of the five algorithms against the default synthetic dataset where k varies from 10 to 50 in Figure 4. As expected, all techniques proposed are effective since the performance of them degrades slowly against the growth of k. The NAVIE and NAIVES algorithms are much slower than the other algorithms, and the I/O costs of them are also much higher than the others. Then for better report on the performance of the algorithms, we exclude the NAIVE and NAIVES algorithms in the following experiments since they have been significantly outperformed by BAS, TKDOM and TKDOM* algorithms.

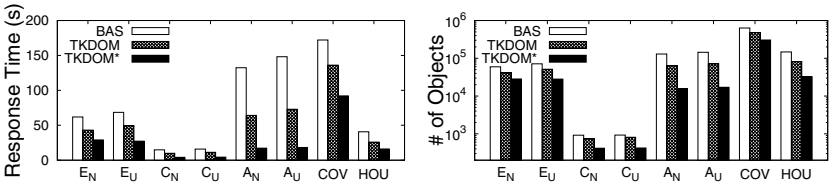
Impact of Data Distribution. We evaluate the performance of BAS, TKDOM and TKDOM* against 8 datasets in Figure 5, where C_N denotes the 3-dimensional synthetic data whose centers and instances follow the Correlated and Normal distributions respectively, and similar definitions go to C_U, E_N, E_U, A_N and A_U. It is observed that the distribution of the instances (N and U) does not noticeably affect performance of the algorithms, so we only perform tests on normal distribution in the following experiments. On the other



(a) *Response Time* (b) *# objects accessed*

Fig. 4. Diff. top k

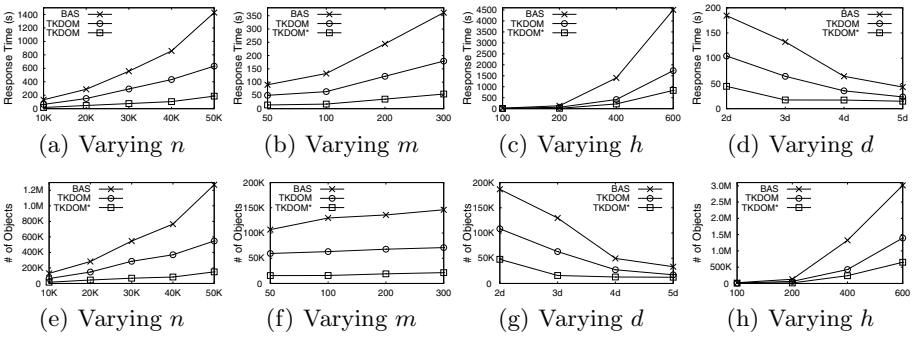
side, all algorithms are very sensitive to the distribution of the object centers. This is because of the nature of dominating query. It is easy to distinguish and sort the *dominance scores* of objects to get a small size of candidate set under the correlated distribution.Whereas anti-correlated distribution leads to more computation time because each object only fully dominate a limited number of other objects, so we use anti-correlated distribution as a default setting for locations. As expected, TKDOM* significantly outperforms other algorithms under all data distributions.



(a) *Response Time* (b) *# objects accessed*

Fig. 5. Impact of Data Distributions

Evaluating Impacts by Different Setting. We study the scalability of our algorithms regarding different number of objects (n) in one dataset, number of instances (m), length of MBR edges (h), and the dimensionality (d) in Figure 6. The response time and the number of I/O increase with the increase of number of objects, instance number, MBR edge length. Clearly, the dataset size increases with objects and instances number thus the filtering processing becomes more expensive. Larger MBR edge length makes it difficult to prune object pairs as there is larger overlap in their MBRs. While the results decrease when d arises. This is because the average number of objects *dominated* or *partially dominated* by each object decreases against the growth of the dimensionality. That means with the increase of dimensionality d, the average area of MBRs gets smaller compared to the whole data space; consequently, the power of the pruning rules becomes more significant. Note that the edge lengths of the objects remains unchanged when the dimensionality grows in the experiments. The results also demonstrate that each pruning rule is very effective and significantly reduces the processing time and I/O cost.



(a) Varying n (b) Varying m (c) Varying h (d) Varying d

(e) Varying n (f) Varying m (g) Varying d (h) Varying h

Fig. 6. Varying Parameters

5 Related Work

With the emergence of many recent important and novel applications involving uncertain data, there has been a great deal of research attention dedicated to this field. Particularly, top k queries are important in analyzing uncertain data. Unlike a top k query over certain data which returns the k best alternatives according to a ranking function, a top k query against uncertain data has inherently more sophisticated semantics. A large amount of work has been dedicated to top k queries with different semantics such as U-Top$k$ and U-kRank [16], PT-k [8], expected rank top k [3], c-Typical-Top$k$ [6] and unified top k [9] semantics. The top k semantics can be widely utilized in various queries, including range query [21], nearest neighbor query [22], reverse nearest neighbor query [2], etc.

Top k dominating queries are introduced by Papadias *et al.* [15] to retrieve the k points that dominate the largest number of other points. Unlike skyline query, the result does not necessarily contain skyline points. Top k dominating queries are widely studied on certain data [18,19,17].

Recently, uncertain query processing has received an increasing attention in many applications. Top k dominating query on uncertain data has been studied in [10,11]. They propose probabilistic top k dominating (PTD) query to retrieve k uncertain objects. Zhang *et al.* [20] formalize threshold-based probabilistic top k dominating queries to overcome some inherent computational deficiency in an exact computation. Nevertheless, none of them supports the *parameterized ranking* semantics. Recently, Feng *et al.* investigate the problem of probabilistic top k dominating query over sliding windows [5]. However, they only consider the x-tuple uncertain model.Techniques proposed in [10,11,20,5] cannot be applied to the top k dominating model proposed in our paper due to the inherent difference of the models.

6 Conclusion

We investigate the problem of identifying top k dominating objects over uncertain data. We formally define a new model for the top k dominating query on multi-dimensional uncertain data. By utilizing the popular R-tree indexing techniques as well as spatial based and rank score based pruning techniques, we develop an effective and efficient algorithm following the *filtering and verification* paradigm. We further improve the performance of the algorithm based on some simple statistics information of the objects. Our experiments convincingly demonstrate the effectiveness and efficiency of our techniques. In the future, we can further investigate the top k dominating query over large dataset.

Acknowledgement. Ying Zhang is supported by ARC DE140100679 and DP130103245. Wenjie Zhang is supported by ARC DE120102144 and DP120104168. Xuemin Lin is supported by NSFC61232006, NSFC61021004, ARC DP120104168 and DP110102937.

References

1. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: ICDE 2001 (2001)
2. Cheema, M.A., Lin, X., Zhang, W., Zhang, Y.: Influence zone: Efficiently processing reverse k nearest neighbors queries. In: ICDE, pp. 577–588 (2011)
3. Cormode, G., Li, F., Yi, K.: Semantics of ranking queries for probabilistic data and expected ranks. In: ICDE (2009)
4. Dalvi, N.N., Suciu, D.: Efficient query evaluation on probabilistic databases. VLDB J. 16(4), 523–544 (2007)
5. Feng, X., Zhao, X., Gao, Y., Zhang, Y.: Probabilistic top-k dominating query over sliding windows. In: Ishikawa, Y., Li, J., Wang, W., Zhang, R., Zhang, W. (eds.) APWeb 2013. LNCS, vol. 7808, pp. 782–793. Springer, Heidelberg (2013)
6. Ge, T., Zdonik, S., Madden, S.: Top-k queries on uncertain data: On score distribution and typical answers. In: SIGMOD (2009)
7. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: SIGMOD Conference, pp. 47–57 (1984)
8. Hua, M., Pei, J., Zhang, W., Lin, X.: Ranking queries on uncertain data: A probabilistic threshold approach. In: SIGMOD (2008)
9. Li, J., Saha, B., Deshpande, A.: A unified approach to ranking in probabilistic databases. VLDB J. 20(2) (2011)
10. Lian, X., Chen, L.: Top-k dominating queries in uncertain databases. In: EDBT, pp. 660–671 (2009)
11. Lian, X., Chen, L.: Probabilistic top-k dominating queries in uncertain databases. Inf. Sci. 226, 23–46 (2013)
12. Lin, X., Zhang, Y., Zhang, W., Cheema, M.A.: Stochastic skyline operator. In: ICDE, pp. 721–732 (2011)
13. Meester, R.: A Natural Introduction to Probability Theory (2004)
14. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Efficient olap operations in spatial data warehouses. In: Jensen, C.S., Schneider, M., Seeger, B., Tsotras, V.J. (eds.) SSTD 2001. LNCS, vol. 2121, pp. 443–459. Springer, Heidelberg (2001)

15. Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive skyline computation in database systems. ACM Trans. Database Syst. 30(1) (2005)
16. Soliman, M.A., Ilyas, I.F., Chang, K.C.: Top-k query processing in uncertain databases. In: ICDE 2007 (2007)
17. Tiakas, E., Papadopoulos, A.N., Manolopoulos, Y.: Progressive processing of subspace dominating queries. VLDB J. 20(6), 921–948 (2011)
18. Yiu, M.L., Mamoulis, N.: Efficient processing of top-k dominating queries on multi-dimensional data. In: VLDB, pp. 483–494 (2007)
19. Yiu, M.L., Mamoulis, N.: Multi-dimensional top-k dominating queries. VLDB J. 18(3), 695–718 (2009)
20. Zhang, W., Lin, X., Zhang, Y., Pei, J., Wang, W.: Threshold-based probabilistic top-k dominating queries. VLDB J. 19(2), 283–305 (2010)
21. Zhang, Y., Lin, X., Tao, Y., Zhang, W.: Uncertain location based range aggregates in a multi-dimensional space. In: ICDE, pp. 1247–1250 (2009)
22. Zhang, Y., Lin, X., Zhu, G., Zhang, W., Lin, Q.: Efficient rank based knn query processing over uncertain data. In: ICDE (2010)