J Intell Inf Syst
DOI 10.1007/s10844-010-0141-4

Probabilistic skylines on uncertain data: model and bounding-pruning-refining methods

Bin Jiang · Jian Pei · Xuemin Lin · Yidong Yuan

Received: 29 November 2009 / Revised: 5 November 2010 / Accepted: 9 November 2010

Abstract Uncertain data are inherent in some important applications. Although a considerable amount of research has been dedicated to modeling uncertain data and answering some types of queries on uncertain data, how to conduct advanced analysis on uncertain data remains an open problem at large. In this paper, we tackle the problem of *skyline analysis on uncertain data*. We propose a novel *probabilistic skyline model* where an uncertain object may take a probability to be in the skyline, and a p-skyline contains all objects whose skyline probabilities are at least p ($0 < p \leq 1$). Computing probabilistic skylines on large uncertain data sets is challenging. We develop a bounding-pruning-refining framework and three algorithms systematically. The bottom-up algorithm computes the skyline probabilities of some selected instances of uncertain objects, and uses those instances to prune other instances and uncertain objects effectively. The top-down algorithm recursively partitions the instances of uncertain objects into subsets, and prunes subsets and objects

This research is supported in part by an NSERC Discovery Grant, an NSERC Discovery Accelerator Supplement Grant, the ARC Discovery Grants (DP110102937, DP0987557, DP0881035), and a Google research Award. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

B. Jiang (✉) · J. Pei
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
e-mail: bjiang@cs.sfu.ca

J. Pei
e-mail: jpei@cs.sfu.ca

X. Lin · Y. Yuan
School of Computer Science and Engineering,
The University of New South Wales and NICTA, Sydney, NSW, Australia

X. Lin
e-mail: lxue@cse.unsw.edu.au

Y. Yuan
e-mail: yyidong@cse.unsw.edu.au

aggressively. Combining the advantages of the bottom-up algorithm and the top-down algorithm, we develop a hybrid algorithm to further improve the performance. Our experimental results on both the real NBA player data set and the benchmark synthetic data sets show that probabilistic skylines are interesting and useful, and our algorithms are efficient on large data sets.

Keywords Uncertain data · Skyline queries · Probabilistic queries · Algorithms

1 Introduction

Uncertain data are inherent in some important applications, such as environmental surveillance, market analysis, and quantitative economics research. Uncertain data in those applications are generally caused by factors like data randomness and incompleteness, limitations of measuring equipment, delayed data updates, etc. Due to the importance of those applications and the rapidly increasing amount of uncertain data collected and accumulated, analyzing large collections of uncertain data has become an important task. Although a considerable amount of research has been dedicated to modeling uncertain data and answering some types of queries on uncertain data (please see Section 7 for a brief review), how to conduct advanced analysis on uncertain data remains an open problem at large. Particularly in this study, we will address the problem of skyline analysis.

1.1 Motivating examples

Many previous studies (e.g., Borzsonyi et al. 2001; Chan et al. 2006a; Huang et al. 2006; Lin et al. 2005; Pei et al. 2005, 2007a; Sharifzadeh and Shahabi 2006; Tao et al. 2006; Yuan et al. 2005) showed that skyline analysis is very useful in multi-criteria decision making applications. As an example, consider analyzing NBA players using multiple technical statistics criteria (e.g., the number of assists and the number of rebounds). Ideally, we want to find the perfect player who can achieve the best performance in all aspects. Such a player, however, does not exist. The skyline analysis here is meaningful since it discloses the tradeoff among the merits of multiple aspects.

A player U is in the skyline if there exists no other player V such that V is better than U in one aspect, and is not worse than U in all other aspects. Skyline analysis on the technical statistics data of NBA players can identify excellent players and their outstanding merits.

We argue that skyline analysis is also meaningful on uncertain data. Consider the skyline analysis on NBA players again. Since the annual statistics are used as certain data in previous studies (Pei et al. 2005), it has never been addressed in the skyline analysis that players may have different performances in different games. *If the game-by-game performance data are considered, which players should be in the skyline and why?*

For example, let us use the number of assists and the number of rebounds, both the larger the better, to examine the players. The two measures may vary substantially player by player and game by game. Uncertainty is inherent due to many factors such as the fluctuations of players' conditions, the locations of the games, and the support from audience. *How can we define the skyline given the uncertain data?*

While a skyline analysis using the real NBA game records will be reported in Section 8, here we plot a few games of five synthetic players in Fig. 1 to illustrate several important issues.

The traditional method represents an attribute of each player using an aggregate function such as the mean or the median, and computes the skyline over such aggregate values. However, such aggregate values cannot capture the performance distribution information, and the skyline computed using such aggregate values may not be meaningful. First, performances in different games may vary differently. For example, in Fig. 1, player Arbor's performances are quite consistent while Eddy's performances are quite diverse. Although Eddy's performance in one game (point *b* in the figure) is better than Arbor's performances in all games in both the number of assists and the number of rebounds, Arbor is generally better than Eddy in the number of assists if all games they played are considered. Second, some outliers may bias the aggregate of a player. For example, Bob is good in general, but he has an outlier game (point *a* in the figure) of poor performance in both measures.

In order to handle the uncertain data, a naïve approach is to compute the skyline on the game records instead of the players. However, the game records can be regarded as the samples of the players' performances and the samples cannot be complete. A skyline game record may be just an exception of a player (e.g., point *b* of Eddy in Fig. 1). Thus, the skyline of game records may not be meaningful for comparing players.

There can be a large number of players over years and each player may play many games in his career. Therefore, the efficiency of skyline analysis on uncertain data matters.

There are many other application examples for skyline analysis on uncertain data. For example, in a digital camera market, one product may receive multiple evaluations which may vary to one another. The total review of the product is not certain and can be modeled as an uncertain object, where each evaluation is regarded as an instance. An evaluation can be multidimensional, including the ratings on price, quality, service, etc. Skyline products can be regarded as good candidates for purchase. The analysis leverages multiple evaluation attributes.

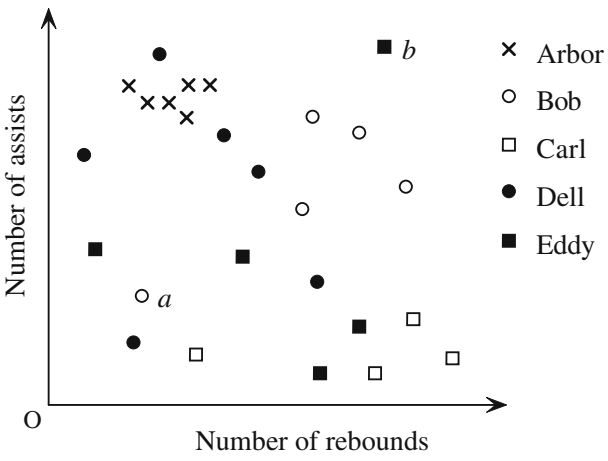


Fig. 1 A set of synthetic players

To evaluate the effect of therapies in medical practice, test cases are collected, and a few measures are used. Generally, the measures may vary, sometimes even substantially, among the test cases of one therapy. Uncertainty is inherent due to the incompleteness of the samples and many other factors (e.g., the physical conditions of patients). Finding the skyline therapies on the uncertain data helps to identify good therapies and understand the tradeoff among multiple factors in question.

As one more example, consider damage control of typhoons (or hurricanes). Tens of thousands of automatic observation stations can be deployed in the area affected by typhoons to collect data like wind intensity and precipitation. A location is likely more seriously damaged if its wind intensity and precipitation are both large under a typhoon attack. However, there are more than ten typhoons every year, and each typhoon takes a different route. Thus, it will be useful to model a location as an uncertain object and the wind intensity and precipitation as the attributes. When a location is affected by a typhoon, the data are recorded as an occurrence of the uncertain object. Based on the data, we can analyze the most likely seriously damaged locations.

In summary, uncertain data pose a few new challenges for skyline analysis and computation. Specifically, we need a meaningful yet simple model for skylines on uncertain data. Moreover, we need to develop efficient algorithms for such skyline computation.

1.2 Challenges and our contributions

In this paper, we address two major challenges about skyline analysis and computation on uncertain data.

1.2.1 Challenge 1: modeling skylines on uncertain data

In a set of uncertain objects, each object has multiple instances, or alternatively, each object is associated with a probability density function. A model about skylines on uncertain data needs to answer two questions:

- How can we capture the dominance relation between uncertain objects?
- What should be the skyline on those uncertain objects?

Our contributions We introduce the probabilistic nature of uncertain objects into the skyline analysis. We follow the possible world model (Abiteboul et al. 1987; Imielinski and Witold Lipski 1984; Sarma et al. 2006) which has been adopted extensively in recent studies on uncertain data processing, such as Soliman et al. (2007), Benjelloun et al. (2006) and Burdick et al. (2005).

Essentially, to compare the advantages between two objects, we calculate the probability that one object dominates the other. Based on the probabilistic dominance relation, we propose the notion of *probabilistic skyline*. The probability of an object being in the skyline is the probability that the object is not dominated by any other objects.

Given a probability threshold p ($0 \leq p \leq 1$), the *p-skyline* is the set of uncertain objects each of which takes a probability of at least p to be in the skyline.

Comparing to the traditional skyline analysis, probabilistic skyline analysis is more informative on uncertain objects.

- First, traditional skylines, computed either using aggregates or individual instances, can be biased by outliers and do not consider the distribution of the instances of an uncertain object. Probabilistic skylines, on the other hand, take all instances of an object and their distribution together to determine the dominance relationship, thus can provide more reliable results.
- Second, the size of the traditional skyline can be large when the data set has a large cardinality or dimensionality (Chan et al. 2006a, c). Users cannot further compare objects in the skyline and have to turn to other analytical tools. This makes the result difficult to process. However, probabilistic skylines can naturally rank objects according to their skyline probabilities. The size of the p-skyline can be controlled by tuning the probability threshold p. This provides a more user-friendly interaction to digest the results.

For example, in a case study (details in Section 8) using the game-by-game technical statistics of 1,313 NBA players in 339,721 games, the traditional skyline computed on average player statistics has 20 players. By contrast, the 0.3-skyline includes five players, the 0.2-skyline includes 14 players, and the 0.1-skyline includes 42 players. Among them, some players that have relatively high skyline probabilities, such as Hakeem Olajuwon (0.204) and Kobe Bryant (0.2), are not in the traditional skyline where only the aggregate statistics are used. On the other hand, some players that are in the traditional skyline have a low skyline probability, such as Gary Payton (0.126) and Lamar Odom (0.102). These are due to their biased game records. We will provide more explanation in Section 8. Clearly, this information cannot be obtained using the traditional skyline analysis.

To the best of our knowledge, we are the first to study skyline analysis on uncertain objects.

1.2.2 Challenge 2: efficient computation of probabilistic skylines

Computing a probabilistic skyline is much more complicated than computing a skyline on certain data. Particularly, in many applications, the probability density function of an uncertain object is often unavailable explicitly. Instead, a set of instances are collected in the hope of approximating the probability density function. According to the possible world model, the probabilistic skyline should be derived from an exponential number of possible worlds. Thus, it is challenging to compute probabilistic skylines on uncertain objects, each of which is represented by a set of instances.

In this paper, we focus on the discrete case of probabilistic skylines computation, i.e., each uncertain object is represented by a set of instances. There are several challenges associated with computing probabilistic skylines in the discrete case. First, each uncertain object may have many instances to be processed. Second, we have to consider many probabilities in deriving the probabilistic skylines. For example, as reported in Section 8, a straightforward method takes more than 1 h to compute the 0.3-skyline on the NBA data set. Using the techniques developed in this paper, we are able to compute probabilistic skylines efficiently and outperform exhaustive search methods by orders of magnitude.

Our contributions We develop a bounding-pruning-refining framework. As the implementation of the framework, we devise three algorithms to tackle the problem.

- The bottom-up algorithm computes the skyline probabilities of some selected instances of uncertain objects, and uses those instances to prune other instances and uncertain objects effectively.
- The top-down algorithm recursively partitions the instances of uncertain objects into subsets, and prunes subsets and objects aggressively.
- Combining the advantages of the bottom-up algorithm and the top-down algorithm, we develop a hybrid algorithm to further improve the performance. We greedily assign objects to the bottom-up method or the top-down method for processing according to the distribution of the instances and the relationship with respect to other objects.

Our methods are efficient and scalable. As verified by our extensive experimental results, our methods are at least one order of magnitude faster than the exhaustive method.

1.2.3 Paper organization

The rest of the paper is organized as follows. In Section 2, we propose the model of probabilistic skylines on uncertain data. In Section 3, we propose the bounding-pruning-refining framework. The bottom-up method and the top-down method are developed in Sections 4 and 5, respectively. We devise the hybrid method in Section 6. We review the related work in Section 7. A systematic performance study is reported in Section 8. We conclude the paper in Section 9.

2 Probabilistic skylines

In this section, we present the probabilistic skyline model. For reference, a summary of notations is given in Table 1.

We first recall the notions of the dominance relation and skylines on certain objects. Then, we extend the dominance relation to probabilistic dominance relation on uncertain objects. Last, we extend the skylines on certain objects to probabilistic skylines on uncertain objects.

Table 1 The summary of notations

Notation	Definition
U, V	Uncertain objects
u, v	Instances of uncertain objects
$\|U\|$	The number of instances of U
f_U	The probability density function of U
p_u	The probability of u to appear
$Pr[U \prec V]$	The probability that U dominates V
$Pr(\cdot)$	Skyline probability of U or u
$Pr^+(\cdot)$	The upper bound of $Pr(U)$ or $Pr(u)$
$Pr^-(\cdot)$	The lower bound of $Pr(U)$ or $Pr(u)$
$U.MBB$	The minimum bounding box of U
$U_{\max}(U_{\min})$	The upper (lower) corner of $U.MBB$

2.1 Skylines on certain objects

By default, we consider points in an n-dimensional numeric space $\mathbf{D} = (D_1, \ldots, D_n)$. The dominance relation is built on the preferences in attributes $D_1, \ldots, D_n$. Without loss of generality, we assume that, on $D_1, \ldots, D_n$, smaller values are more preferable.

For two points u and v, u is said to *dominate* v, denoted by $u \prec v$, if for every dimension D_i $(1 \leq i \leq n)$, $u.D_i \leq v.D_i$, and there exists a dimension D_{i_0} $(1 \leq i_0 \leq n)$ such that $u.D_{i_0} < v.D_{i_0}$.

Given a set of points S, a point $u \in S$ is a *skyline point* if there exists no other point $v \in S$ such that v dominates u. The *skyline* on S is the set of all skyline points.

Example 1 (Dominance and skyline) Consider the points in Fig. 2. According to the definition of dominance, point c dominates b, d, and e. Points a, c and f are not dominated by any other points in the set. Thus, these three points form the skyline of this data set.

2.2 Probabilistic skylines

An *uncertain object* is conceptually described by a *probability density function* (PDF) f in the data space $\mathbf{D}$. Generally, $f(u) \geq 0$ for any point u in the data space $\mathbf{D}$, and $\int_{u \in \mathbf{D}} f(u)\, du = 1$. This is referred to as the *continuous* case.

Practically, the probability density function of an uncertain object is often unavailable explicitly. Instead, an uncertain object U is represented by a set of *instances* (points) such that each instance $u \in U$ has a probability p_u to appear. Such a representation, referred to as the *discrete* case, correspondingly has the property that $0 < p_u \leq 1$ and $\sum_{u \in U} p_u = 1$.

To keep our model simple, we assume that *uncertain objects are independent*. That is, an instance of an object does not depend on the instances of any other objects. Moreover, we assume that, *for an uncertain object, each instance carries the same probability to happen*. Although the rest of this paper bears the above two assumptions, our model can be easily extended to cases where dependencies (e.g., correlations or anti-correlations) exist among objects and instances carry different weights.

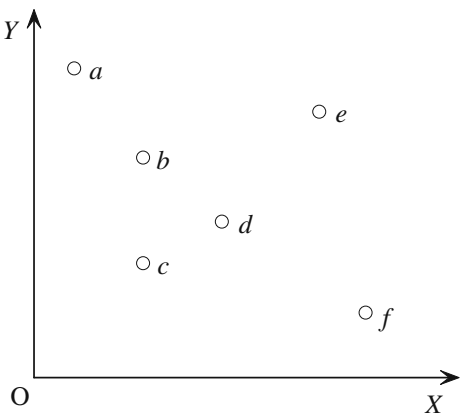


Fig. 2 A set of certain points

Now, let us extend the dominance relation to uncertain objects, and show how this can straightforwardly define skylines on uncertain objects.

Let U and V be two uncertain objects, and f_U and f_V be the corresponding probability density functions, respectively. Then, the probability that V dominates U is

$$\begin{aligned} Pr[V \prec U] &= \int_{u \in D} f_U(u) \left(\int_{v \prec u} f_V(v)\, \mathrm{d}v \right) \mathrm{d}u \\ &= \int_{u \in D} \int_{v \prec u} f_U(u)\, f_V(v)\, \mathrm{d}v \mathrm{d}u \end{aligned} \tag{1}$$

In the discrete case, the probability that V dominates U is given by

$$Pr[V \prec U] = \sum_{u \in U} p_u \cdot \left(\sum_{v \in V, v \prec u} p_v \right) \tag{2}$$

Since any two points u and v in the data space must have one of the following three relations: $u \prec v$, $v \prec u$, or u and v do not dominate each other, for two uncertain objects U and V, $Pr[U \prec V] + Pr[V \prec U] \leq 1$.

Example 2 (Probabilistic dominance relation) Consider the set of three uncertain objects in Fig. 3. Each object has two instances. Assume each instance takes equal probability to appear, that is, the appearance probability of each instance is 0.5.

For instances of object C, c_1 is dominated by every instance of A, and c_2 is dominated by instance a_2 of A. Thus, the probability that A dominates C is $Pr[A \prec C] = 0.5 \times 1 + 0.5 \times 0.5 = 0.75$. Similarly, we can calculate $Pr[B \prec C] = 0.5$.

Since c_1 is dominated by every instance of object A and c_2 is dominated by every instance of object B, the probability that C is dominated by A or B is 1. In other words, C cannot be in the skyline.

Because $Pr[A \prec C]$ and $Pr[B \prec C]$ are not independent, an important observation here is that, although $Pr[A \prec C] = 0.75 < 1$ and $Pr[B \prec C] = 0.5 < 1$, the probability of C being dominated by A or B is 1. Moreover, $Pr[(A \not\prec C) \wedge (B \not\prec C)] \neq (1 - Pr[A \prec C]) \cdot (1 - Pr[B \prec C])$.

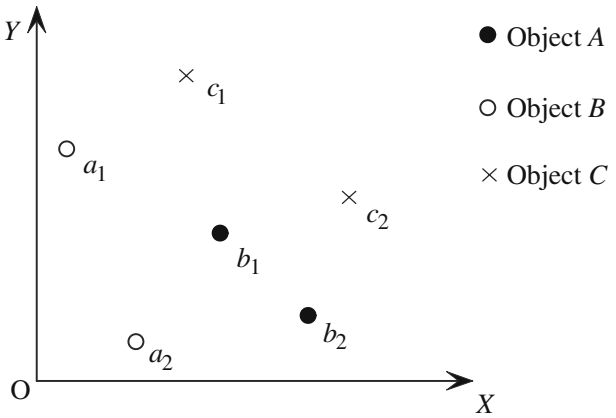


Fig. 3 A set of uncertain objects

The observation in Example 2 indicates that *the probabilistic dominance relations are not independent and cannot be used straightforwardly to define skylines on uncertain objects*. Then, what is the probability that an uncertain object is in the skyline?

We first illustrate our probabilistic skyline model in the discrete case. Then we show the model in the continuous case.

Given a set of uncertain objects $\mathbf{S} = \{U_1, \cdots, U_m\}$, a *possible world* $\mathbf{w} = \{u_1, \cdots, u_m\}$ is a set of m instances such that each uncertain object in $\mathbf{S}$ has one instance in $\mathbf{w}$. The probability of $\mathbf{w}$ to appear is

$$Pr(\mathbf{w}) = \prod_{i=1}^{m} p_{u_i}.$$

Let Ω be the set of all possible worlds, then

$$\sum_{\mathbf{w} \in \Omega} Pr(\mathbf{w}) = 1.$$

Let $Sky(\mathbf{w})$ denote the set of objects such that for every object $U \in Sky(\mathbf{w})$, the instance of U in $\mathbf{w}$ is in the skyline of $\mathbf{w}$. Then, the probability that U appears in the skylines of the possible worlds is

$$Pr(U) = \sum_{U \in Sky(\mathbf{w}), \mathbf{w} \in \Omega} Pr(\mathbf{w}).$$

$Pr(U)$ is called the *skyline probability* of U.

Example 3 (Probabilistic skylines) Consider the set of uncertain objects in Fig. 3 again. We have eight possible worlds in total. Each possible world has the probability $0.5^3 = 0.125$ to appear.

$P(A) = 1$ since a_1 and a_2 are in the skyline of every possible world. Moreover, $P(C) = 0$ because c_1 and c_2 are not in the skyline of any possible world.

Note that B is in the skylines of four possible worlds $\{a_1, b_1, c_1\}$, $\{a_1, b_1, c_2\}$, $\{a_1, b_2, c_1\}$, and $\{a_1, b_2, c_2\}$. Therefore, $P(B) = 4 \times 0.125 = 0.5$.

For each instance u of $U \in \mathbf{S}$, $Pr(u)$, the probability of u being in the skyline, is

$$Pr(u) = \prod_{V \in \mathbf{S} \setminus \{U\}} \left(1 - \sum_{v \in V, v \prec u} p_v\right). \tag{3}$$

$Pr(u)$ is called the *skyline probability* of instance u.

By the above definition, it can be immediately verified that

$$Pr(U) = \sum_{u \in U} p_u \cdot Pr(u). \tag{4}$$

Consequently, we have

$$P(U) = \sum_{u \in U} p_u \cdot Pr(u) = \sum_{u \in U} \left(p_u \cdot \prod_{V \in \mathbf{S} \setminus \{U\}} \left(1 - \sum_{v \in V, v \prec u} p_v\right)\right). \tag{5}$$

Similarly, in the continuous case, the skyline probability $Pr(U)$ is defined as

$$Pr(U) = \int_{u \in \mathbf{D}} f_U(u) \prod_{V \neq U} \left(1 - \int_{v \prec u} f_V(v)\, \mathrm{d}v\right) \mathrm{d}u. \quad (6)$$

An uncertain object may take a probability to be in the skyline. It is natural to extend the notion of skyline to *probabilistic skyline*. For a set of uncertain objects **S** and a *probability threshold* p ($0 \leq p \leq 1$), the *p-skyline* is the subset of objects in **S** each of which takes a probability of at least p to be in the skyline. That is,

$$Sky(p) = \{U \in \mathbf{S} | Pr(U) \geq p\}.$$

Problem Definition 1 Given a set of uncertain objects **S** and a probability threshold p ($0 \leq p \leq 1$), the problem of *probabilistic skyline computation* is to compute the p-skyline on **S**.

Particularly, in this paper, we tackle the discrete case problem. That is, given a set of uncertain objects where each object is a set of sample instances and a probability threshold p, compute the p-skyline.

Although we will focus on the discrete case in this paper, some of our ideas can be applied to handle the continuous case, which will be discussed briefly in Section 9. Moreover, we only address the exact algorithms in this paper. The development of approximation algorithms for probabilistic skylines is very interesting and is investigated systematically in another study we are conducting.

3 The bounding-pruning-refining framework

On a large uncertain data set, the number of possible worlds can be huge. For example, consider a data set of 1,000 uncertain objects. If each uncertain object has four instances, the number of possible worlds $|\Omega| = 4^{1,000} > 10^{602}$. It is impractical to compute the skylines in all possible worlds one by one and derive the skyline probability for each uncertain object.

To tackle the problem, we propose a bounding-pruning-refining framework. A probabilistic skyline computation method can conduct iterations in the following three steps.

Bounding — For an uncertain object U, we try to obtain an upper bound and a lower bound on the skyline probability of U. This can be achieved by, for example, computing the skyline probabilities of some selected instances of U, or partitioning U into some subsets where the skyline probability of each subset can be bounded.

Pruning — For an uncertain object U, if the lower bound of $Pr(U)$ is larger than or equal to p, the probability threshold, then U is in the p-skyline. If the upper bound of $Pr(U)$ is smaller than p, then U is not in the p-skyline. In both cases, we do not need to compute the skyline probabilities of instances in U anymore.

Refining — If p is between the lower bound and the upper bound of $Pr(U)$, then we need to get tighter bounds of the skyline probabilities by the next iteration of bounding, pruning and refining.

The above iteration goes on until for every uncertain object we can determine whether it is in the p-skyline or not.

In the next two sections, we will propose two methods implementing the above bounding-pruning-refining framework. The two methods differ in how to compute and refine the bounds and how to prune uncertain objects. The bottom-up method is described in Section 4 and the top-down method is presented in Section 5.

4 The bottom-up method

In the bottom-up method, in the bounding step, we compute the skyline probabilities of a small subset of instances. In the pruning step, an uncertain object may be pruned using the skyline probabilities of its instances, or those of some other objects. The method is called *bottom-up* since the bound computation and refinement start from the instance level (bottom) and go up to the object level.

4.1 Bounding skyline probabilities

Given an uncertain object U and an instance u of U, trivially, we have $0 \leq Pr(U) \leq 1$ and $0 \leq Pr(u) \leq 1$. Let

$$U_{\min} = \left(\min_{i=1}^{|U|} \{u_i.D_1\}, \ldots, \min_{i=1}^{|U|} \{u_i.D_n\} \right) \text{ and}$$

$$U_{\max} = \left(\max_{i=1}^{|U|} \{u_i.D_1\}, \ldots, \max_{i=1}^{|U|} \{u_i.D_n\} \right)$$

be the *minimum and the maximum corners* of the minimum bounding box (MBB for short) of U, respectively. Note that, $U_{\min}$ and $U_{\max}$ are not necessary two actual instances of U. In this case, we treat them as virtual instances and define their skyline probabilities following (3). That is,

$$Pr(U_{\min}) = \prod_{V \neq U} \left(1 - \frac{|\{v \in V \mid v \prec U_{\min}\}|}{|V|} \right), \text{ and}$$

$$Pr(U_{\max}) = \prod_{V \neq U} \left(1 - \frac{|\{v \in V \mid v \prec U_{\max}\}|}{|V|} \right).$$

Lemma 1 (Bounding skyline probabilities) *Let $U = \{u_1, \ldots, u_l\}$ be an uncertain object where $u_1, \ldots, u_l$ are the instances of U.*

1. *If $u_{i_1} \prec u_{i_2}$ $(0 \leq i_1, i_2 \leq l)$, then $Pr(u_{i_1}) \geq Pr(u_{i_2})$.*
2. *$Pr(U_{\min}) \geq Pr(U) \geq Pr(U_{\max})$.*

Proof The dominance relation on instances is transitive: for instances x, y, and z of an uncertain object, if $x \prec y$ and $y \prec z$, then $x \prec z$. Since $u_{i_1} \prec u_{i_2}$, for any instance v of other object V, if $v \prec u_{i_1}$ then $v \prec u_{i_2}$.

Applying the transitivity to (3), we have

$$Pr(u_{i_1}) = \prod_{V \neq U} \left(1 - \frac{|\{v \in V \mid v \prec u_{i_1}\}|}{|V|}\right) \geq \prod_{V \neq U} \left(1 - \frac{|\{v \in V \mid v \prec u_{i_2}\}|}{|V|}\right) = Pr(u_{i_2})$$

The first item in the lemma is proved.

According to item 1 in this lemma, for any u_i $(1 \leq i \leq l)$, $Pr(U_{\min}) \geq Pr(u_i) \geq Pr(U_{\max})$. Item 2 in the lemma follows with the above inequality and (4). □

Lemma 1 provides a means to compute the upper bounds and the lower bounds of instances and uncertain objects using the skyline probabilities of other instances.

According to the first inequality in the lemma, the skyline probability of an instance can be bounded by those of other instances dominating or dominated by it. In other words, when the skyline probability of an instance is calculated, the bounds of the skyline probabilities of some other instances of the same object may be refined accordingly.

The second inequality in the lemma indicates that the minimum and the maximum corners of the MBB can play important roles in bounding the skyline probability of a set of instances.

4.2 Pruning techniques

If the skyline probability of an uncertain object or an instance of an uncertain object is computed, can we use this information to prune the other uncertain instances or objects? Following with Lemma 1, we immediately have the following rule to determine the p-skyline membership of an uncertain object using its minimum or maximum corners.

Pruning Rule 1 *For an uncertain object U and probability threshold p, if $Pr(U_{\min}) < p$, then U is not in the p-skyline. If $Pr(U_{\max}) \geq p$, then U is in the p-skyline.*

Moreover, we can prune an uncertain object using the upper bounds and the lower bounds of the skyline probabilities of instances.

Pruning Rule 2 *Let U be an uncertain object. For each instance $u \in U$, let $Pr^+(u)$ and $Pr^-(u)$ be the upper bound and the lower bound of $Pr(u)$, respectively. If $\frac{1}{|U|}\sum_{u \in U} Pr^+(u) < p$, then U is not in the p-skyline. If $\frac{1}{|U|}\sum_{u \in U} Pr^-(u) \geq p$, then U is in the p-skyline.*

We can also use the information about one uncertain object to prune other uncertain instances or objects. First, if an instance u of an uncertain object U is dominated by the maximum corner of another uncertain object V, then u can never be in the skyline in any possible world. Figure 4 illustrates this pruning rule.

Pruning Rule 3 *Let U and V be uncertain objects such that $U \neq V$. If u is an instance of U and $V_{\max} \prec u$, then $Pr(u) = 0$.*

By pruning some instances in an uncertain object using the above rule, we can reduce the cost of computing the skyline probability of the object.

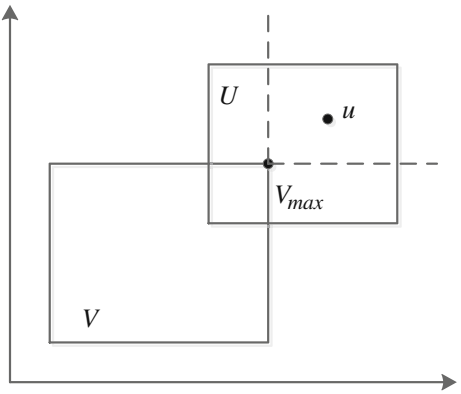


Fig. 4 An illustration of Pruning Rule 3

When the skyline probabilities of some instances of an uncertain object are computed, we can use the information to prune some other uncertain objects.

Pruning Rule 4 *Let U and V be two uncertain objects and $U' \subseteq U$ be a subset of instances of U such that $U'_{\max} \preceq V_{\min}$. If*

$$\frac{|U - U'|}{|U|} \cdot \min_{u \in U'}\{Pr(u)\} < p,$$

then $Pr(V) < p$ and thus V is not in the p-skyline.

Proof Figure 5 illustrates the situation. Since $V_{\min}$ is dominated by all instances in U'. An instance of V can be in the skyline only if U does not appear as any instance in U'. Even if no instance in $(U - U')$ dominates any instance of V, the probability that V is in the skyline still cannot reach the probability threshold p, since $U'_{\max} \preceq V_{\min}$ and $\frac{|U-U'|}{|U|} \cdot \min_{u \in U'}\{Pr(u)\} < p$. Thus V cannot be in the p-skyline.

Formally, since every instance of V is dominated by all instances in U', only when U takes an instance in $(U - U')$, V may have a chance of not being dominated by U. The probability that an instance of V is not dominated by an instance of U cannot be

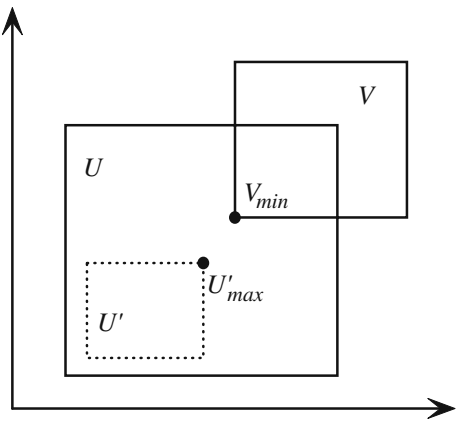


Fig. 5 An illustration of Pruning Rule 4

more than $(1 - \frac{|U'|}{|U|}) = \frac{|U-U'|}{|U|}$. Moreover, since $U'_{\max} \preceq V_{\min}$, all instances of objects other than U and V dominating $U'_{\max}$ also dominate $V_{\min}$.

Thus,

$$Pr(V) \leq Pr(V_{\min}) \leq \left(1 - \frac{|U'|}{|U|}\right) \cdot Pr(U'_{\max}) = \frac{|U - U'|}{U} \cdot Pr(U'_{\max})$$
$$\leq \frac{|U - U'|}{U} \cdot \min_{u \in U'}\{Pr(u)\} < p$$

V is not in the p-skyline. □

As a special case of Pruning Rule 4, if there exists an instance $u \in U$ such that $Pr(u) < p$ and $u \preceq V_{\min}$, then $Pr(V) < p$ and V can be pruned.

The pruning rule is powerful since even an uncertain object partially computed can be used to prune other objects.

4.3 Refinement strategies

For an uncertain object U, we want to determine whether U is in the p-skyline by computing the skyline probabilities of as few instances of U as possible. Finding an optimal subset of instances to compute is a very difficult online problem since, without computing the probabilities of the instances, we do not know their distribution. Here, we propose a layer-by-layer heuristic method.

4.3.1 Layers of instances

According to the first inequality in Lemma 1, among all instances of an object U, we can first compute the skyline probabilities of the instances that are not dominated by any other instances, i.e., the skyline instances in the object. Those instances are the *layer*-1 instances, as illustrated in Fig. 6. The skyline probabilities of the instances at *layer*-1 can serve as the upper bounds of the skyline probabilities of other instances, and generate an upper bound of the skyline probability of U.

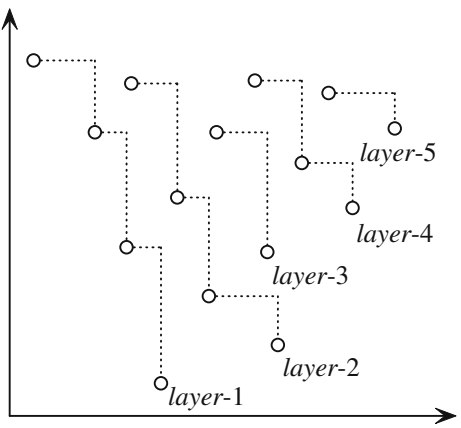


Fig. 6 The layers of an uncertain object

If the upper bounds using the *layer*-1 instances are not enough to qualify or disqualify U in the p-skyline, we need to refine the upper bounds. We can compute the skyline probabilities of the instances at *layer*-2 which are dominated only by the instances at *layer*-1, as shown in Fig. 6, too. Similarly, we can partition the instances of an object into layers.

Formally, for an uncertain object U, an instance $u \in U$ is at *layer*-1 if u is not dominated by any other instance in U. An instance v is at *layer-k* ($k > 1$) if, v is not at the 1st, ..., $(k-1)$-th layers, and v is not dominated by any instances except for those at the 1st, ..., $(k-1)$-th layers.

The advantage of partitioning instances of an object into layers is that, once the skyline probabilities of all instances at one layer are calculated, the probabilities can be used as the upper bounds of the instances at the higher layers.

Lemma 2 *In an uncertain object U, let $u_{1,1}, \ldots, u_{1,l_1}$ be the instances at layer-k_1, and $u_{2,1}, \ldots, u_{2,l_2}$ be the instances at layer-k_2, $k_1 < k_2$. Then, for any instance at layer-k_2 u_{2,j_2} ($1 \le j_2 \le l_2$), there exists an instance at layer-k_1 u_{1,j_1} ($1 \le j_1 \le l_1$) such that $Pr(u_{1,j_1}) \ge Pr(u_{2,j_2})$. Moreover, $\max_{i=1}^{l_1}\{Pr(u_{1,i})\} \ge \max_{j=1}^{l_2}\{Pr(u_{2,j})\}$.*

Proof Since $k_1 < k_2$, instance u_{2,j_2} must be dominated by an instance at *layer-k_1*. Otherwise, u_{2,j_2} is at *layer-k_1* or some lower layer. Let u_{1,j_1} be an instance at *layer-k_1* that dominates u_{2,j_2}. Then, the first inequality follows with Lemma 1. The second inequality follows with the first inequality. □

4.3.2 Partitioning instances to layers

How can instances of an object be assigned quickly into layers?

For each instance u, we define the *key* of the instance as the sum of its values in all attributes, that is, $u.key = \sum_{i=1}^{n} u.D_i$. Then, we sort all the instances in the key ascending order. This is motivated by the SFS algorithm (Chomicki et al. 2003). The sorted list of instances has a nice property: for instances u and v such that $u \prec v$, u precedes v in the sorted list.

We scan the sorted list once. The first instance has the minimum key value, and is assigned to *layer*-1. We compare the second instance with the first one. If the second one is dominated, then it is assigned to *layer*-2; otherwise it is assigned to *layer*-1.

Generally, when we process an instance u, suppose at the time there already exist h layers. We compare u with the instances currently at *layer*-$\lceil \frac{h}{2} \rceil$. One of the two cases may happen.

- If u is dominated by an instance at that layer, then u must be at some layer higher than $\lceil \frac{h}{2} \rceil$.
- Otherwise, u is neither dominated by, nor dominates, any instance at that layer. Then, u must be at that layer or some lower layer.

We conduct this binary search recursively until u is assigned to a layer.

Lemma 1 indicates that the minimum corner of the MBB of an uncertain object leads to the upper bounds of the skyline probabilities of all instances as well as the

object itself. As a special case, we assign this minimum corner as a virtual instance at *layer*-0.

The above partitioning method has a nice property: all instances at a layer are sorted in the key ascending order.

4.3.3 Scheduling objects

From which objects should we start the skyline probability computation?

In order to use the pruning rules discussed in Section 4.2 as much as possible, those instances in uncertain objects that likely dominate many other objects or instances should be computed early. Heuristically, those instances which are close to the origin may have a better chance to dominate other objects and instances.

The instances of an uncertain object are processed layer by layer. Within each layer, the instances are processed in the key ascending order. As discussed in Section 4.2, some pruning rules enable us to use the partial information of some uncertain objects to prune other objects and instances, we interleave the processing of different objects.

Technically, all instances of an uncertain object are kept in a list. The minimum corner of its MBB is treated as a special instance and put at the head of the list. The heads of lists of all uncertain objects are organized into a heap. We iteratively process the top instance in the heap. If an object cannot be pruned after its minimum corner is processed, we organize the rest of instances in its list in the layer and key value ascending order. Once an instance from an object is processed, the object sends the next instance into the heap if its skyline membership is not determined. The proper pruning rules are triggered if the conditions are satisfied.

4.4 Algorithm and implementation

The bottom-up algorithm is shown in Fig. 7. We explain some critical implementation details here.

4.4.1 Finding possible dominating objects

For an object U, we want to find all the other objects that may contain some instances dominating U. Those objects are called the *possible dominating objects* of U. The skyline membership of U depends on only those possible dominating objects. All other objects that do not contain any instances dominating U do not need to be considered.

To speed up the search of possible dominating objects, we employ R-trees (Guttman 1984). An R-tree is a tree data structure for indexing multidimensional data, such as points, etc. A node of an R-tree contains a set of entries. Each entry at a leaf node is in the form of $\langle pID, coords\rangle$ where pID refers to the point ID and *coords* is the coordinates of the point. Each entry in a non-leaf node is in the form of $\langle child, child.MBB\rangle$ where *child* refers to a child node, and $child.MBB$ is the minimum bounding box of all entries in this child node. Approximately, an R-tree can be built in time $O(n \log n)$ where n is the number of points indexed. A range query can be answered in time $O(\log n)$.

We organize the minimum corners of MBBs of all objects into a global R-tree. To find the possible dominating objects of U, we issue a window query with the

Fig. 7 The bottom-up algorithm

Algorithm *BottomUp*(**S**, p)
Input: a set of uncertain objects **S**; probability threshold p;
Output: the p-skyline in **S**;
Method:
1: **SKY** $= \emptyset$;
2: FOR EACH object $U \in$ **S** DO
3: $Pr^+(U) = 1$; $Pr^-(U) = 0$;
4: compute U_{min}, the minimum corner of its MBB;
END FOR EACH
5: build an R-tree to store U_{min} for all $U \in$ **S**;
6: build a heap H on U_{min} for all $U \in$ **S**;
7: WHILE $H \neq \emptyset$ DO
8: let $u \in U$ be the top instance in H;
9: IF u is from a non-skyline object THEN NEXT;
10: IF u is dominated by another object THEN
11: GOTO Line 20; // Pruning Rule 3
12: IF u is the minimum corner of U THEN
13: find possible dominating objects of U; // Section 4.1.1
14: compute $Pr(u)$; // Section 4.4.2
15: IF $Pr(u) \geq p$ THEN
16: partition instances of U to layers; // Section 4.3.2
17: ELSE U is pruned; // Pruning Rule 1
ELSE
18: compute $Pr(u)$; // Section 4.4.2
19: $Pr^-(U) = Pr^-(U) + \frac{1}{|U|} Pr(u)$;
20: IF u is the last instance at a layer THEN
21: update $U.Pr_{max}$;
22: $Pr^+(U) = Pr^-(U) + U.Pr_{max} \cdot \frac{|\widetilde{U}|}{|U|}$;
23: IF $\frac{|U-U'|}{|U|} \cdot \min_{u \in U'}\{Pr(u)\} < p$ THEN
24: apply Pruning Rule 4 to prune other objects;
25: IF $Pr^-(U) \geq p$ THEN
26: **SKY** $=$ **SKY** $\cup \{U\}$; NEXT; // Pruning Rule 2
27: IF $Pr^+(U) \geq p$ THEN
28: insert the next instance of U into H;
END WHILE
29: RETURN **SKY**;

origin and $U_{\max}$ as the opposite corners on the global R-tree. The possible dominating objects for an object are computed only when the minimum corner of the object is popped from the heap.

If an object U does not have any possible dominating objects, then every instance of U is not dominated by any instance of other objects. In other words, the skyline probability of U is 1.

4.4.2 Computing skyline probability

To compute the skyline probability $Pr(u)$ for an instance $u \in U$, we compare u with the possible dominating objects of U one by one. To facilitate the comparison, we incrementally maintain a local R-tree T_V for each object V. T_V is set to empty in the initialization. When an instance $u \in U$ is compared with object V, we insert into T_V the instances in V that have a key value less than $u.key$, since only those instances

in V may dominate u. Then, we issue a window query with the origin and u as the opposite corners to compute $|\{v \in V | v \prec u\}|$.

After comparing u with all possible dominating objects of U, using (4), we can calculate $Pr(u)$. We also update the lower bound of the probability of object U immediately as

$$Pr^-(U) = Pr^-(U) + \frac{1}{|U|} Pr(u).$$

Once all instances in a layer are processed, as discussed in Lemma 2, we use the maximum probability of instances in this layer as the upper bound (denoted by $U.Pr_{\max}$) of the probabilities of instances in the higher layers. Moreover, the upper bound of the probability of U is updated as

$$Pr^+(U) = Pr^-(U) + U.Pr_{\max} \cdot \frac{|\widetilde{U}|}{|U|},$$

where $\widetilde{U} \subseteq U$ is the set of instances whose probabilities are not calculated yet.

4.4.3 Using Pruning Rule 4

In order to use Pruning Rule 4 to prune other objects, for each object U, we maintain U' as the set of instances which precede the current processing instance in its instance list. The skyline probability of those instances are already computed. Once U' satisfies the condition in the rule, we compute $U'_{\max}$, the maximum corner of the MBB of U', and issue a window query on the global R-tree described in Section 4.4.1 with $U'_{\max}$ and the maximum corner of the MBB of all objects in the data set as the opposite corners. For each minimum corner returned from this query, the corresponding uncertain object satisfies the pruning rule and thus is not in the p-skyline. We note that for each object, this rule is applied at most once. This is because once this condition is satisfied, it will be always satisfied afterwards.

4.5 Cost analysis of the bottom-up algorithm

It can be immediately verified that the cost of the bottom-up algorithm is predominated by computing the skyline probabilities of instances as presented in Section 4.4.2. Suppose that R is the average cost of querying the local R-trees of possible dominating objects, with all pruning techniques are applied, for computing the skyline probabilities of instances. Let W_{total} denote the number of instances whose skyline probabilities are computed in the algorithm. Then, the average cost of the algorithm is $O(W_{\text{total}} \cdot R)$.

As shown in our experimental results, in practice many instances and objects can be pruned sharply. The bottom-up algorithm only has to compute a small portion of instances. That is, W_{total} is much smaller than the total number of instances. Thus, the bottom-up algorithm has good scalability on the large data sets used in our experiments.

5 The top-down method

In this section, we present a top-down method for probabilistic skyline computation. The method starts with the whole set of instances of an uncertain object. The skyline probability of the object can be bounded using the maximum and the minimum corners of the MBB of the object. To improve the bounds, we can recursively partition the instances into subsets. The skyline probability of each subset can be bounded using its MBB in the same way. Facilitated by (4), the skyline probability of the uncertain object can be bounded as the weighted mean of the bounds of subsets. Once the p-skyline membership of the uncertain object is determined, the recursive bounding process stops.

5.1 Partition trees

To facilitate the partitioning process, we use a *partition tree* data structure for each uncertain object. A partition tree is binary. Each leaf node contains a set of instances and the corresponding MBB. Each internal node maintains the MBB of all instances in its descendants and the total number of instances.

The construction of a partition tree for an uncertain object is somewhat similar to that of kd-trees (Bentley 1975). We start with a tree of only one node—the root node which contains all instances of the object and the MBB. The tree grows in rounds. In each round, a leaf node with l instances ($l > 1$) is partitioned into two children nodes according to one attribute such that the left child and the right child contain $\lceil \frac{l}{2} \rceil$ and $\lfloor \frac{l}{2} \rfloor$ instances, respectively.

We take a simple round robin method to choose the attributes to grow a partition tree. The attributes are sorted into $D_1, \ldots, D_n$ in an arbitrary order. The root node (level-0) is partitioned into two children in attribute D_1, those children (level-1) are split into grand-children in attribute D_2, and so on. To split the nodes at level-n, attribute D_1 is used again.

The time complexity to grow one level of the tree for an uncertain object U is $O(|U|)$. The cost to fully grow a partition tree (i.e., each leaf node contains only one instance) is $O(|U| \log_2 |U|)$ since the tree has at most $\log_2 |U|$ levels.

5.2 Bounding using partition trees

For a node N in a partition tree, we also use N to denote the set of instances allocated to N. Let $N.MBB$ be the MBB of the instances allocated to N, and $N_{\max}$ and $N_{\min}$ be the maximum and the minimum corners, respectively. Then, by Lemma 1, for any instance $u \in N$, the skyline probability of u can be bounded by

$$Pr(N_{\max}) \leq Pr(u) \leq Pr(N_{\min}). \tag{7}$$

Moreover, if the partition tree of uncertain object U has l leaf nodes $N_1, \ldots, N_l$, then

$$\frac{1}{|U|} \sum_{i=1}^{l} |N_i| \cdot Pr(N_{i,\max}) \leq Pr(U) \leq \frac{1}{|U|} \sum_{i=1}^{l} |N_i| \cdot Pr(N_{i,\min}), \tag{8}$$

where $N_{i,\max}$ and $N_{i,\min}$ are the maximum and the minimum corners of $N_i.MBB$, respectively, and $|N_i|$ is the number of instances in N_i.

Computing the exact skyline probabilities for all corners can be costly. Instead, we estimate the bounds. To bound the skyline probabilities for $N_{\min}$ and $N_{\max}$ for a node N in the partition tree of uncertain object U, we query the possible dominating objects of U as described in Section 4.4.1. We traverse the partition tree of each possible dominating object V of U in the depth-first manner. When a node M in the partition tree of V is met, one of the following three cases happens.

- If $M_{\max}$ dominates $N_{\min}$ (as shown in Fig. 8a), then $N_{\min}$ and $N_{\max}$ are dominated by all instances in M. That is, $Pr(N_{\min}) \leq Pr(M_{\max})$.
- If $M_{\min}$ does not dominate $N_{\max}$ (as shown in Fig. 8b), then no instance in M can dominate either $N_{\min}$ or $N_{\max}$.
- If the above two situations do not happen, then some instances in M may dominate some instances in N (as shown in Fig. 8c). If M is an internal node, we traverse the left and the right children of M recursively. Otherwise, M is a leaf node. Then, we estimate a lower bound of $Pr(N_{\max})$ by assuming all instances in M dominate $N_{\max}$, and an upper bound of $Pr(N_{\min})$ by assuming no instance in M dominates $N_{\min}$.

By traversing all partition trees of the possible dominating objects, we apply (3) to compute the upper bound for $Pr(N_{\min})$ and the lower bound for $Pr(N_{\max})$. With the two bounds and inequality (8), we can immediately bound the skyline probability of object U. We use only the maximum and the minimum corners of the MBBs, and never compute the skyline probability of any one in a subset of instances.

5.3 Pruning and refinement using partition trees

When one level is grown for the partition trees of all uncertain objects whose skyline memberships are not determined, the possible dominating objects of them are also partitioned to the same level. For all new leaf nodes grown in this round, we bound their probabilities by traversing the partition trees of the corresponding possible dominating objects. We note that the computation of such bounding for the leaf nodes which have the same MBB can be shared.

After that, we check whether some uncertain objects or some leaf nodes in some partition trees may be pruned. That is, their skyline probabilities do not need to be computed any more. Pruning those nodes can make the skyline computation faster.

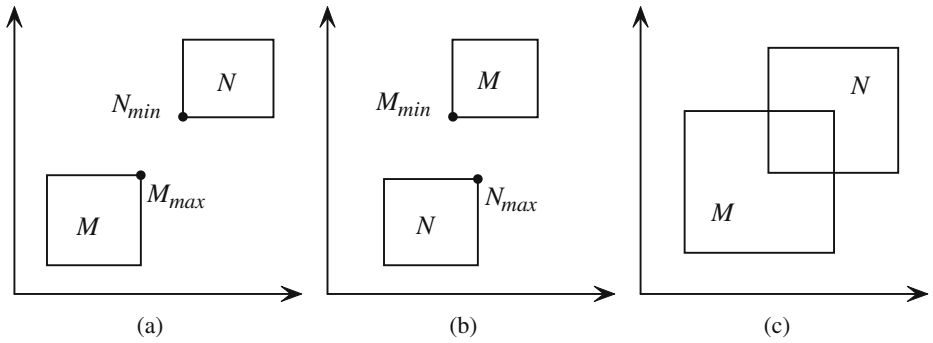


Fig. 8 Three cases of bounding $Pr(N)$

Consider a node N in the partition tree of uncertain object U. If there exists another uncertain object $V \neq U$ such that $N_{\min}$ is dominated by $V_{\max}$, then any instance in N cannot be in the skyline. In other words, $Pr(u) = 0$ for any $u \in N$. We do not need to compute any subset of N anymore since the instances there cannot contribute to the skyline probability of U. Figure 9 illustrates this pruning rule.

Pruning Rule 5 *Let N be a node in the partition tree of uncertain object U. If there exists an object $V \neq U$ such that $V_{\max} \preceq N_{\min}$, then node N can be pruned.*

Moreover, if $Pr(N_{\min}) = Pr(N_{\max})$, according to inequality (7), the skyline probability of any instance in N is determined. N can be pruned.

Pruning Rule 6 *Let N be a node in the partition tree of uncertain object U. If $Pr(N_{\min}) = Pr(N_{\max})$, then for each $u \in N$, $Pr(u) = Pr(N_{\min}) = Pr(N_{\max})$ and node N can be pruned.*

Last, once the skyline probability of an uncertain object can be bounded at least p or less than p, then whether the object is in the p-skyline is determined. We do not need to refine the estimation of the probability of this object anymore.

Pruning Rule 7 *Let p be the probability threshold. Suppose the partition tree of an uncertain object U has l leaf nodes $N_1, \ldots, N_l$. Let $N_{i,\max}$ and $N_{i,\min}$ be the maximum and the minimum corners of $N_i.MBB$, respectively.*

- *If* $\frac{1}{|U|}\sum_{i=1}^{l} |N_i| \cdot Pr(N_{i,\max}) \geq p$, *then U is in the p-skyline.*
- *If* $\frac{1}{|U|}\sum_{i=1}^{l} |N_i| \cdot Pr(N_{i,\min}) < p$, *then U is not in the p-skyline.*

In both cases, the partition tree of U can be pruned.

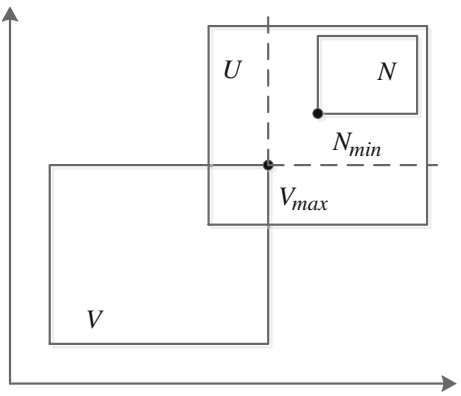


Fig. 9 An illustration of Pruning Rule 5

After the pruning step using the above rules, only the partition trees of those uncertain objects which cannot be determined in the p-skyline or not are left. In such trees, only those nodes whose skyline probabilities are not determined survive.

In a refinement step, we partition those surviving leaf nodes and their possible dominating objects to one more level. With the refinement, the bounds of skyline probabilities are tighter.

5.4 The top-down algorithm and cost analysis

The top-down algorithm is shown in Fig. 10. In the implementation, we use an R-tree to index the minimum corners of the MBBs of all objects so that the search of possible dominating objects can be conducted efficiently.

Let P be the average cost of querying partition trees of possible dominating objects for bounding the skyline probabilities of the minimum and maximum corners of MBBs, and M_{total} be the number of tree nodes whose skyline probabilities are bounded in the algorithm. Then, the average cost of the algorithm is $O(M_{\text{total}} \cdot P)$.

As will be shown in our experimental results, many nodes can be pruned sharply by the pruning rules. The top-down algorithm only has to grow a small number of

Fig. 10 The top-down algorithm

Algorithm $Topdown(S, p)$
Input: a set of uncertain objects **S**; probability threshold p;
Output: the p-skyline in **S**;
Method:
1: **SKY** $= \emptyset$;
2: FOR EACH object $U \in$ **S** DO
3: initialize a partition tree T_U with only the root node;
END FOR EACH
4: let L be the set of all partition trees; $i = 0$;
5: WHILE $L \neq \emptyset$ DO
6: FOR EACH partition tree $T_U \in L$ DO
7: FOR EACH leaf node N of level-i in T_U DO
8: bound $Pr(N_{min})$ and $Pr(N_{max})$; // Section 5.2
9: bound $Pr(U)$; // Inequality (8)
10: IF $Pr(U) \geq p$ THEN
11: **SKY** $=$ **SKY** $\cup \{U\}$; $L = L - \{T_U\}$; NEXT;
12: ELSE IF $Pr(U) < p$ THEN
13: $L = L - \{T_U\}$; NEXT; // Pruning Rule 7
ELSE
14: apply Pruning Rules 5 and 6 to N if applicable;
15: partition N to level-$(i+1)$ if it cannot be pruned;
END FOR EACH
END FOR EACH
16: $i = i + 1$;
END WHILE
17: RETURN **SKY**;

tree nodes (i.e., M_{total} is small), and has good scalability with respect to cardinality of the data sets.

6 The hybrid method

In this section, we develop a hybrid method combining the advantages of the bottom-up method and the top-down method. The general idea is to partition the set $\mathbf{S}$ of uncertain objects into two subsets $\mathbf{S}_{BU}$ and $\mathbf{S}_{TD}$ such that they likely can be processed by the bottom-up method and the top-down method efficiently, respectively.

In Section 6.1, we analyze the advantages and the disadvantages of the bottom-up method and the top-down method, and present the framework of the hybrid method. In Section 6.2, we discuss how to partition uncertain objects into two subsets $\mathbf{S}_{BU}$ and $\mathbf{S}_{TD}$. In Section 6.3, we apply the layer structure in the bottom-up method to improve the top-down method.

6.1 The framework of the hybrid method

In our bounding-pruning-refining framework, we refine the upper bound and the lower bound of the skyline probability of every uncertain object. If the upper bound of the skyline probability of an uncertain object is less than p, the probability threshold, the object is not in the p-skyline and thus can be pruned. If the lower bound of the skyline probability of an uncertain object is at least p, the object is in the p-skyline and can be removed from further refinement. Interestingly, the bottom-up method and the top-down method have different edges in bounding the skyline probabilities of uncertain objects and pruning them.

The bottom-up method is good at pruning non-skyline objects. A non-skyline object has a skyline probability smaller than the probability threshold. For a non-skyline object U, the bottom-up method can quickly obtain a tight upper bound of $Pr(U)$ using the layer structure, since the instances in U having large skyline probabilities are processed before those having small skyline probabilities. Once the upper bound of the skyline probability of U is determined lower than the probability threshold, U can be pruned.

However, for a skyline object U', the lower bound of $Pr(U')$ may not increase fast in the bottom-up method, since the lower bound is the sum of the skyline probabilities of the instances of U' processed so far. For example, if the probability threshold $p = 0.9$ and every instance in an uncertain object U' takes the same probability to appear, in the bottom-up method, we have to process at least 90% of the instances of U' before the lower bound can be at least 0.9. In such a situation, U' cannot be determined early. However, the top-down algorithm is good at determining the skyline membership of skyline objects quickly.

In the top-down method, the recursive partitioning isolating the instances of low skyline probabilities quickly and thus the lower bound of the skyline probability of an uncertain object can be estimated tighter and quicker than the bottom-up method.

Based on the above discussion, we propose a hybrid method. Using a method will be given in Section 6.2, we quickly estimate whether an uncertain object may be in the skyline. For the subset of uncertain objects $\mathbf{S}_{TD}$ whose estimations are positive (i.e., the objects are likely in the skyline), we use the top-down method. For the subset

Fig. 11 The hybrid algorithm

Algorithm $Hybrid(S, p)$
Input: a set of uncertain objects **S**; probability threshold p;
Output: the p-skyline in **S**;
Method:
1: $\mathbf{S}_{BU} = \emptyset$; $\mathbf{S}_{TD} = \emptyset$;
2: FOR EACH object $U \in S$ DO
3: IF $Pr_{est}(U) < p$ THEN $\mathbf{S}_{BU} = S_{BU} \cup \{U\}$
ELSE $\mathbf{S}_{TD} = S_{TD} \cup \{U\}$
END FOR EACH
4: RETURN $BottomUp(S_{BU}, p) \cup$
$ImprovedTopDown(S_{TD}, p)$;

of uncertain objects $\mathbf{S}_{BU}$ whose estimations are negative (i.e., the objects are likely not in the skyline), we apply the bottom-up method. The framework of the hybrid method is shown in Fig. 11.

6.2 Estimating skyline probability

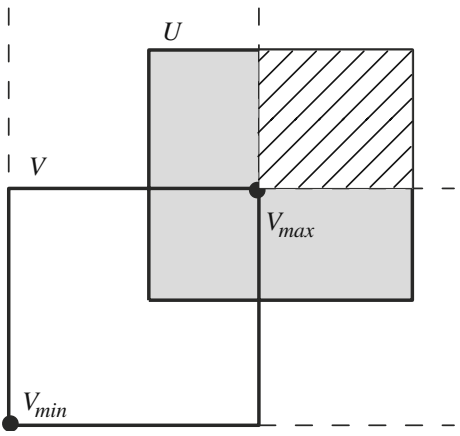
How can we quickly estimate whether an uncertain object has a good chance to be in the p-skyline? The skyline probability of an uncertain object U depends on two aspects.

- *Instance distribution:* the distribution of the instances of U; and
- *Uncertain object distribution:* the distribution of the instances of other uncertain objects.

To quickly estimate the skyline probability of an object, we assume that the instances of an uncertain object are uniformly distributed within its MBB, so that we can approximate the estimation using the MBBs of uncertain objects.

For an uncertain object U, let V be a possible dominating object of U. Let $DA_{V_{\min}}(U)$ denote the area of U dominated by the minimum corners of V, i.e., the gray area in Fig. 12. Moreover, let $DA_{V_{\max}}(U)$ denote the area of U dominated by the maximum corner of V, i.e., the shaded area in Fig. 12. Then, under the assumption

Fig. 12 Estimate skyline probability



that the instances in U and V are uniformly distributed in their MBBs, we estimate the probability of V dominating U as

$$Pr_{\text{est}}[V \prec U] = \frac{DA_{V_{\min}}(U) + DA_{V_{\max}}(U)}{2 \cdot area(U)} \tag{9}$$

where $area(U)$ is the area of the MBB of U.

We can see that the larger $DA_{V_{\min}}(U)$ and $DA_{V_{\max}}(U)$, the larger the probability of V dominating U. As an extreme case, when $V_{\max}$ dominates $U_{\min}$, that is, every instance of V dominates all instances of U, $DA_{V_{\min}}(U) = DA_{V_{\max}}(U) = area(U)$, then $Pr_{\text{est}}[V \prec U] = 1$. And when $V_{\min}$ does not dominate $U_{\max}$, $DA_{V_{\min}}(U) = DA_{V_{\max}}(U) = 0$, then $Pr_{\text{est}}[V \prec U] = 0$.

Let $PDO(U)$ denote the set of possible dominating objects of U. Then, the estimated skyline probability of U is the product of the estimated probability of every $V \in PDO(U)$ not dominating U. That is,

$$Pr_{\text{est}}(U) = \prod_{V \in PDO(U)} (1 - Pr_{\text{est}}[V \prec U]). \tag{10}$$

In the hybrid algorithm (Step 3 in Fig. 11), for each uncertain object U, we estimate its skyline probability using the above method. If the estimated skyline probability is less than the probability threshold p, then U is assigned to subset $\mathbf{S}_{BU}$ which will be processed by the bottom-up method. Otherwise, U is assigned to subset $\mathbf{S}_{TD}$ which will be processed by the top-down method.

6.3 Improving the top-down method using the layer structure

We can improve the top-down method by applying the layer structure developed in the bottom-up method (Section 4.3) to obtain a good processing order of leaf nodes in every iteration of partitioning.

In the top-down method (Fig. 10), for each object U in an iteration, we grow one level of the partition tree of U, and bound the skyline probabilities of the maximum and the minimum corners of every leaf node to obtain the upper bound and the lower bound of $Pr(U)$. Any arbitrary order can be used in the top-down method to process the leaf nodes. Fortunately, we can schedule the leaf nodes in a good order so that the efficiency can be improved. Although we cannot obtain an optimal order without knowing the skyline probabilities of the instances, the layer structure developed in the bottom-up method provides a heuristically good processing order.

For a partition tree, we define the *key* of a node N as the sum of all attribute values of $N_{\min}$, i.e., $\sum_{i=1}^{n} N_{\min}.D_i$. Then, we partition all leaf nodes into layers in the same way as we partition the instances of an uncertain object in the bottom-up method (Section 4.3.2). We have the following result based on Lemma 2.

Proposition 1 *In a partition tree of an uncertain object U, let $N_{1,1}, \ldots, N_{1,l_1}$ be the leaf nodes at layer-k_1, $N_{2,1}, \ldots, N_{2,l_2}$ be the leaf nodes at layer-k_2, and $k_1 < k_2$. Then,*

for any leaf node at layer-k_2 N_{2,j_2} $(1 \leq j_2 \leq l_2)$, there exists a leaf node at layer-k_1 N_{1,j_1} $(1 \leq j_1 \leq l_1)$ such that $Pr(N_{1,j_1\min}) \geq Pr(N_{2,j_2\min})$, where $N_{1,j_1\min}$ and $N_{2,j_2\min}$ are the minimum corners of the MBBs of N_{1,j_1} and N_{2,j_2}, respectively. Moreover, $\max_{i=1}^{l_1}\{Pr(N_{1,i\min})\} \geq \max_{j=1}^{l_2}\{Pr(N_{2,j\min})\}$.

According to Proposition 1, we process the leaf nodes of the partition tree of every uncertain object U layer by layer so that the upper bound and the lower bound of $Pr(U)$ both approach the actual value of $Pr(U)$ quickly. Using Proposition 1, we can estimate the upper bound of $Pr(N_{\min})$ for an unprocessed leaf node N. Thus, after a leaf node is processed, we can obtain a tighter upper bound of $Pr(U)$ according to (8). Moreover, heuristically, the skyline probabilities of leaf nodes decrease as the layer number increases. When we process the leaf nodes in the layer increasing order, the lower bound of $Pr(U)$ also increases in a faster pace heuristically.

7 Related work

A preliminary version of this paper appeared as Pei et al. (2007b), which is the first paper to explore skyline analysis on uncertain data. Our study is related to the previous work on querying uncertain data and skyline computation. In this section, we review the major existing results in these two aspects and also the work of Pei et al. (2007b) on probabilistic skyline computation.

7.1 Querying uncertain spatial data

In statistics, there are a number of tools dealing with uncertain and probabilistic data, such as graphical models including Bayesian networks, Markov Random Fields, Influence Diagrams, etc. (Deshpande and Sarawagi 2007). Graphical models present an option for representing the uncertainty in the data and evaluating queries over uncertain data (Dalvi and Suciu 2007; Sen et al. 2007). Particularly, they are useful to model dependence between objects. In this paper, we assume that objects are independent to each other. For future work, we would like to explore graphical models to handle more complex relationship between objects in computing probabilistic skylines.

Modeling and querying uncertain data have also attracted considerable attention from the database research community (see Aggarwal and Yu 2007; Sarma et al. 2006; Dalvi and Suciu 2004, and the references therein). The work that relates closest to our problem is management and query processing of uncertain data in spatial-temporal databases (Cheng et al. 2003, 2004; Tao et al. 2005; Dai et al. 2005; Kriegel et al. 2006).

Cheng et al. (2003) proposed a broad classification of probabilistic queries over uncertain data, and developed novel techniques for evaluating probabilistic queries. Cheng et al. (2004) are the first to study probabilistic range queries. They developed two auxiliary index structures to support querying uncertain intervals effectively. Tao et al. (2005) investigated probabilistic range queries on multi-dimensional space with arbitrary probability density functions. They identified and formulated several

pruning rules and proposed a new access method to optimize both I/O cost and CPU time. Dai et al. (2005) introduced an interesting concept of *ranking* probabilistic spatial queries on uncertain data which selects the objects with highest probabilities to qualify the spatial predicates. On the uncertain data indexed by R-tree, several efficient algorithms were developed to support ranking probabilistic range queries and nearest neighbor queries. Kriegel et al. (2006) proposed to use probabilistic distance functions to measure the similarity between uncertain objects. They presented both the theoretical foundation and some effective pruning techniques of probabilistic similarity joins.

Different from the previous work on querying uncertain spatial data, our study introduces skyline queries and analysis to uncertain data. As shown in Sections 1 and 8, skyline queries are meaningful for uncertain data and can disclose some interesting knowledge that cannot be identified by the existing queries on uncertain data.

7.2 Skyline computation and analysis

Computing skylines was first investigated by Kung et al. (1975) in computational geometry. Bentley et al. (1978) proposed an efficient algorithm with an expected linear runtime if the data distribution on each dimension is independent.

Borzsonyi et al. (2001) introduced the concept of skylines in the context of databases and proposed a SQL syntax for skyline queries. They also developed the skyline computation techniques based on *block-nested-loop* and *divide-and-conquer* paradigms, respectively. Chomicki et al. (2003) proposed another block-nested-loop based computation technique, SFS (*sort-filter-skyline*), to take the advantages of pre-sorting. The SFS algorithm was further significantly improved by Godfrey et al. (2005).

The first *progressive* technique that can output skyline points without scanning the whole dataset was delveloped by Tan et al. (2001). Kossmann et al. (2002) presented another progressive algorithm based on the nearest neighbor search technique, which adopts a divide-and-conquer paradigm on the dataset. Papadias et al. (2003) proposed a *branch-and-bound* algorithm (BBS) to progressively output skyline points on datasets indexed by an R-tree. One of the most important properties of BBS is that it minimizes the I/O cost.

Variations of skyline computation have been explored. Pei et al. (2005, 2007a) and Yuan et al. (2005) proposed a skyline cube data structure that completely precomputes the skylines of all possible subspaces for a given data set. Xia and Zhang (2006) addressed the incremental maintenance of skyline cubes. Tao et al. (2006) developed the SUBSKY algorithm to answer subspace skyline queries efficiently in any subspaces. To tackle the problem of skylines in high dimensional spaces, Chan et al. (2006b) relaxed the notion of dominance to k-dominance and proposed k-dominant skylines. Dellis and Seeger (2007) proposed the reverse skyline query, which consists of objects whose dynamic skyline contains a given query point q. The dynamic skyline of an object p corresponds to a transformed data space where p becomes the origin and all other points are represented by their distance vectors to p. Denis Mindolin (2009) investigated skylines in a case where some attributes are considered to be more important than the others. Lin et al. (2005), Tao and Papadias (2006), and Morse et al. (2006) answered skyline queries over data streams, where

the skyline keeps updating as new data elements come and old data elements expire. Sarma et al. (2009) developed a randomized skyline algorithm for streaming. Jiang and Pei (2009) applied skyline analysis on time series data, where every data object is a time series. Balke et al. (2004) and Wu et al. (2006) computed skylines in distributed systems. Park et al. (2009) computed skyline on multicore architectures. Huang et al. (2006) computed skyline on mobile lightweight devices such as MANETs. Sharifzadeh and Shahabi (2006) proposed spatial skyline queries, where a dimension of a data point is the distance to some query point. Chan et al. (2005) and Sacharidis et al. (2009) considered skyline computation in partially ordered domains. Chen and Lian (2008) considered skyline queries in metric spaces. Zhang et al. (2009b) worked on skyline maintenance to handle frequent updates of the data set. Zhang et al. (2009c) estimated the skyline cardinality based on density estimation. Wong et al. (2007) and Jiang et al. (2008) used skylines to mine user preferences and make recommendations.

All of the studies on skyline computation and analysis reviewed above focus on certain data. Our study extends the skyline computation and analysis to uncertain data. As shown in the previous section, extending skyline queries to uncertain data is far from straightforward. It involves both the development of skyline models and the design of novel algorithms for efficient computation.

7.3 Probabilistic skyline computation on uncertain data

After our preliminary work (Pei et al. 2007b) introduced the concept of probabilistic skyline on uncertain data, there are some following studies adopting our probabilistic skyline model (Section 2).

Lian and Chen (2008) studied the bichromatic probabilistic reverse skyline (BPRS) queries over uncertain data. A BPRS query takes two data sets A, B and a query object q as the input, and outputs those objects $o \in A$ such that the dynamic skyline of o in the data set B contains q. The dynamic skyline of an object p corresponds to a transformed data space where p becomes the origin and all other points are represented by their distance vectors to p. The main techniques to answer a BPRS query also follow our bounding-pruning-refining framework.

Atallah and Qi (2009) proposed to compute the skyline probability of all instances of all objects without setting a threshold. They developed algorithms based on a space partitioning technique. The worst-case time complexity is sub-quadratic to the number of objects, however, exponential to the dimensionality. So, their algorithms mainly focus on the 2-dimensional case, and it is not practical for higher dimensional cases.

Böhm et al. (2009) studied the continuous case of the probabilistic skyline query where every object is modeled as a mixture Gaussian distribution. Their techniques cannot be applied directly to the discrete case.

Zhang et al. (2009a) extended the probabilistic skyline operator to data streams using a sliding window model. Their techniques are also under the bounding-pruning-refining framework with an index structure to efficiently handle updates.

The above studies investigated the variations of our p-skyline query in different environments under the probabilistic skyline model. The techniques developed within either follow the bounding-pruning-refining framework or cannot be applied directly to answer p-skyline queries.

8 Empirical study

In this section, we report an extensive empirical study to examine the effectiveness and the efficiency of probabilistic skyline analysis on uncertain data. All the experiments were conducted on a PC with Intel P4 3.0 GHz CPU and 2 GB main memory running the Debian Linux operating system. All algorithms were implemented in C++.

8.1 Effectiveness of probabilistic skylines

To verify the effectiveness of probabilistic skylines on uncertain data, we use a real data set of the NBA game-by-game technical statistics from 1991 to 2005 downloaded from www.NBA.com. The NBA data set contains 339,721 records about 1,313 players. We treat each player as an uncertain object and the records of the player as the instances of the object. Three attributes are selected in our analysis: number of points, number of assists, and number of rebounds. The larger those attribute values, the better.

Table 2 shows the 0.1-skyline players in the skyline probability descending order. We also conducted the traditional skyline analysis. We calculated the average statistics for each player in each attribute. That is, each player has only one record in the aggregate data set. We computed the skyline on the aggregate data set, which is called the *aggregate skyline* for short hereafter. All skyline players in the aggregate skyline are annotated by a "*" sign in Table 2. We obtain several interesting observations.

Table 2 0.1-skyline players in skyline probability descending order

Name	Skyline probability	Name	Skyline probability
LeBron James*	0.350699	Magic Johnson*	0.151813
Dennis Rodman*	0.327592	Chris Paul*	0.149264
Shaquille O'Neal*	0.323401	Gilbert Arenas	0.142883
Charles Barkley*	0.309311	Clyde Drexler	0.138993
Kevin Garnett*	0.302531	Patrick Ewing	0.13577
Jason Kidd*	0.293569	Rod Strickland	0.135735
Allen Iverson*	0.269871	Brad Daugherty	0.133572
Michael Jordan*	0.250633	Steve Francis	0.131061
Tim Duncan*	0.241252	Dirk Nowitzki	0.130301
Karl Malone*	0.239737	Paul Pierce	0.127079
Chris Webber*	0.22153	Gary Payton*	0.126328
Kevin Johnson*	0.208991	Baron Davis	0.125298
Hakeem Olajuwon	0.203641	Vince Carter	0.122946
Kobe Bryant	0.200272	Antoine Walker	0.121745
Dwyane Wade	0.199065	Steve Nash	0.115874
Tracy Mcgrady	0.198185	Andre Miller	0.11275
Grant Hill*	0.191164	Isiah Thomas	0.11076
John Stockton*	0.183591	Elton Brand	0.10966
David Robinson	0.177437	Scottie Pippen	0.108941
Stephon Marbury*	0.16683	Dominique Wilkins	0.104323
Tim Hardaway*	0.166206	Lamar Odom*	0.101803

First, probabilistic skylines can capture the knowledge obtained from traditional skyline analysis. The top-12 players with the largest skyline probabilities are also in the aggregate skyline. All of them are great players. Those players not only have good average performance so that they are in the aggregate skyline, but also performed outstandingly in some games so that they have a high skyline probability.

Second, traditional skylines can be biased by outliers. Some players that are not in the aggregate skyline may still have a high skyline probability. There are 22 players who are not in the aggregate skyline, but have a higher skyline probability than Odom who is a skyline player in the aggregate data set. Olajuwon is an example. Figure 13 plots the number of points and the number of rebounds of the game records of Olajuwon and Odom at a sample rate of 5% (so that the figure is readable). We observe that Olajuwon has some bad games (e.g, zero point and three rebounds) which hinder his average statistics. On the other hand, Odom has a few good games (e.g, more than 25 points and 12 rebounds) which help his average statistics. But overall, Olajuwon could be a better player since he has many more great games (e.g., 40 points and 19 rebounds).

In more details, Olajuwon is dominated by four other players in the aggregate data set: O'Neal, Barkley, Duncan, and Webber. In Fig. 14, we plot their game records. Olajuwon has some records (e.g., 40 points and 19 rebounds) dominating most records of other players. On the other hand, he also has some records (e.g., zero point and three rebounds) that are dominated by many records of other players. Comparing to the four players dominating him, Olajuwon's performance has a sparser distribution.

Comparing to the aggregate skyline, the probabilistic skyline finds not only players consistently performing well, but also outstanding players with relatively inconsistent performance possibly due to aging or injuries.

Third, a player A may have a higher skyline probability than a player B who dominates A in the aggregate data set. As an example, Ewing has a higher skyline probability than Brand, though Ewing is dominated by Brand in the aggregate data set. We plot a sample with ratio 5% of both players in Fig. 15. Their aggregate values are also shown in the figure. The performance of Ewing is more diverse than that

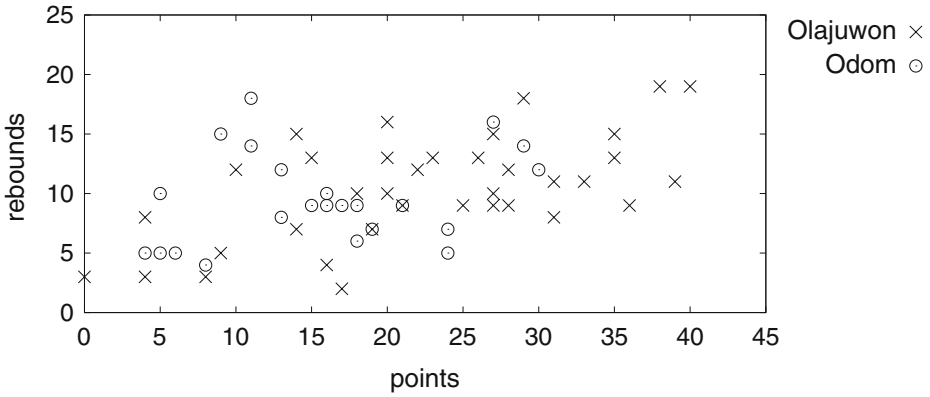


Fig. 13 Comparing Olajuwon's and Odom's records

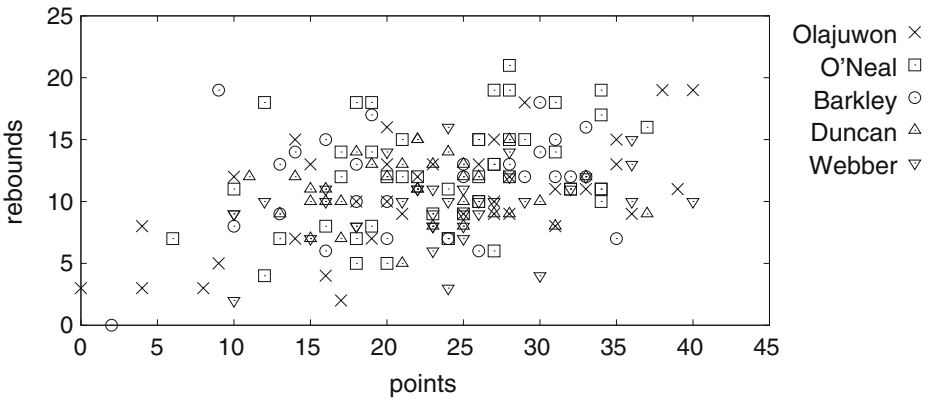


Fig. 14 Olajuwon's and some other players' records

of Brand. Ewing played very well in a few games, which explains why Ewing has a higher skyline probability.

In summary, probabilistic skylines disclose more knowledge about uncertain data by considering the instance distributions of objects which cannot be captured by traditional skyline analysis, and provide a more comprehensive view on advantages of uncertain objects than skylines using only the aggregate of such objects. Interestingly, we can rank uncertain objects using skyline probabilities, while the skyline on aggregate of uncertain data cannot reflect the differences on the opportunities of uncertain objects not to be dominated by other objects. This is another significant advantage of probabilistic skyline analysis.

8.2 Performance evaluation

To verify the efficiency and the scalability of our algorithms, we use the NBA real data set as well as synthetic data sets in anti-correlated, independent, and correlated

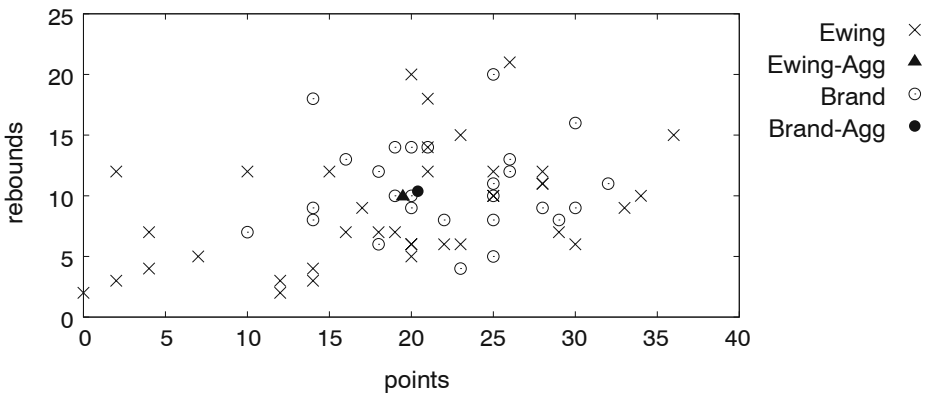


Fig. 15 Ewing's and Brand's records

distributions. For the synthetic data sets, the domain of each dimension is [0, 1]. The dimensionality d by default is 4. The cardinality (i.e., number of uncertain objects) m by default is 10,000. We first generated the centers of all uncertain objects using the benchmark data generator described in Borzsonyi et al. (2001). Then, for each uncertain object, we use the center to generate a hyper-rectangle region where the instances of the object appear. The edge size of the hyper-rectangle region follows a normal distribution in range [0, 0.2] with expectation 0.1 and standard deviation 0.025. The instances of the object distributed uniformly in the region. The number of instances of an uncertain object follows uniform distribution in range $[1, l]$, where l is 400 by default. Thus, in expectation, each object has $\frac{l}{2}$ instances, and the total number of instances in a data set is $\frac{ml}{2}$ (2,000,000 by default). The probability threshold p is 0.3 unless otherwise specified. Table 3 summarizes the experiment settings.

8.2.1 Probabilistic skyline size

Figure 16 shows the size of probabilistic skylines (i.e., the number of objects in a probabilistic skyline) with respect to three important factors: the probability threshold, the dimensionality and the cardinality. Generally, anti-correlated data sets have the largest skyline size. Correlated data sets have the smallest skyline size. This is similar to the situations of skylines on certain objects. As shown in Fig. 16a, the higher the probability threshold, the smaller the skyline size. This is because a p-skyline contains a p'-skyline if $p < p'$. Figure 16b shows the results on the NBA data set, which is in a consistent trend. Figure 16c and d show that the skyline size increases with higher dimensionality and larger cardinality, which is also similar to the situations of skylines on certain data sets. As the dimensionality increases, the data set becomes sparser. An object has a better opportunity not to be dominated in all dimensions. As the cardinality increases, more objects may have chances not to be dominated.

8.2.2 Efficiency and scalability

Figure 17 shows the overall performance of the bottom-up algorithm (BU), the top-down algorithm (TD), the hybrid algorithm (HY), and an exhaustive algorithm (EX) for benchmarking purpose. To compute the p-skyline on a data set, without any pruning techniques, EX has to compute the skyline probability for each uncertain object. The numbers on the bars give the exact runtime of the algorithms on the data sets.

BU, TD, and HY are much faster than EX. The results clearly indicate that the pruning techniques in BU and TD significantly save the cost of computing the exact skyline probabilities of many instances and objects. HY is the fastest on anti-correlated and independent data sets while it is a little slower than BU on the NBA data set.

Table 3 The summary of experiment settings

Notation	Definition (default values)
m	Cardinality of the data set (10,000)
d	Dimensionality of the data set (4)
l	Maximum number of instances per object (400)
p	Probability threshold (0.3)

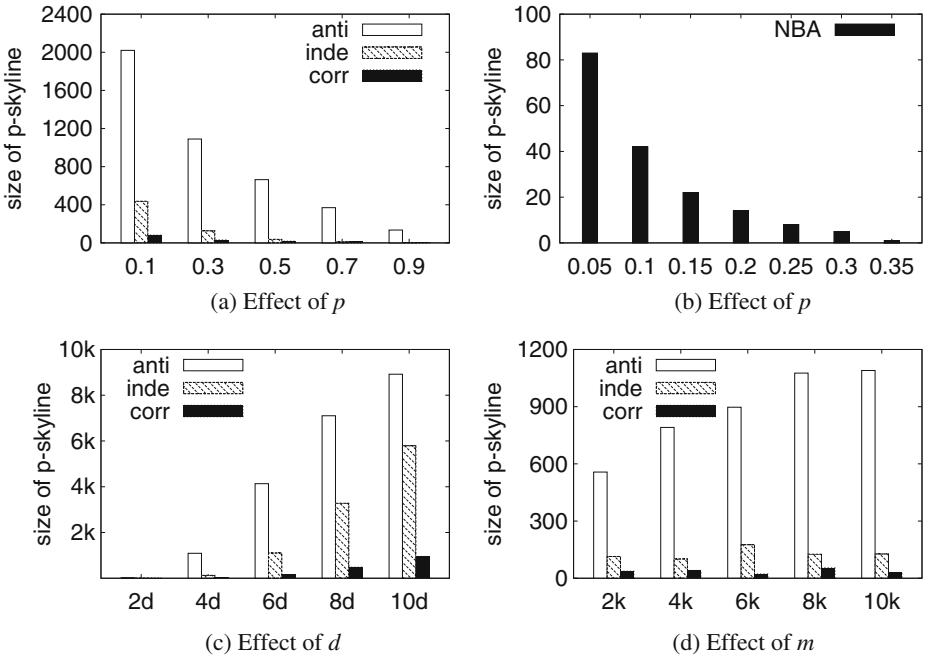
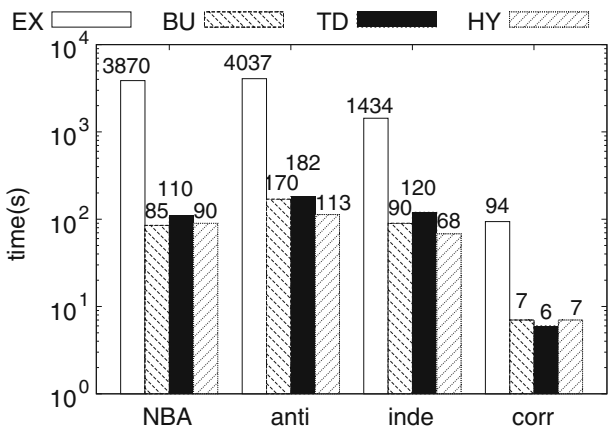


Fig. 16 The size of p-skyline with respect to probability threshold p, dimensionality d, and cardinality m

Computing skylines on anti-correlated data sets is much more challenging than the other cases as reflected the runtime in Fig. 17. In the rest of this section, we focus on analyzing in detail the performance of our algorithms on anti-correlated data sets.

Figure 18 compares BU, TD, and HY with respect to probability threshold, dimensionality, and cardinality. All algorithms follow similar trends. The hybrid

Fig. 17 Overall performance



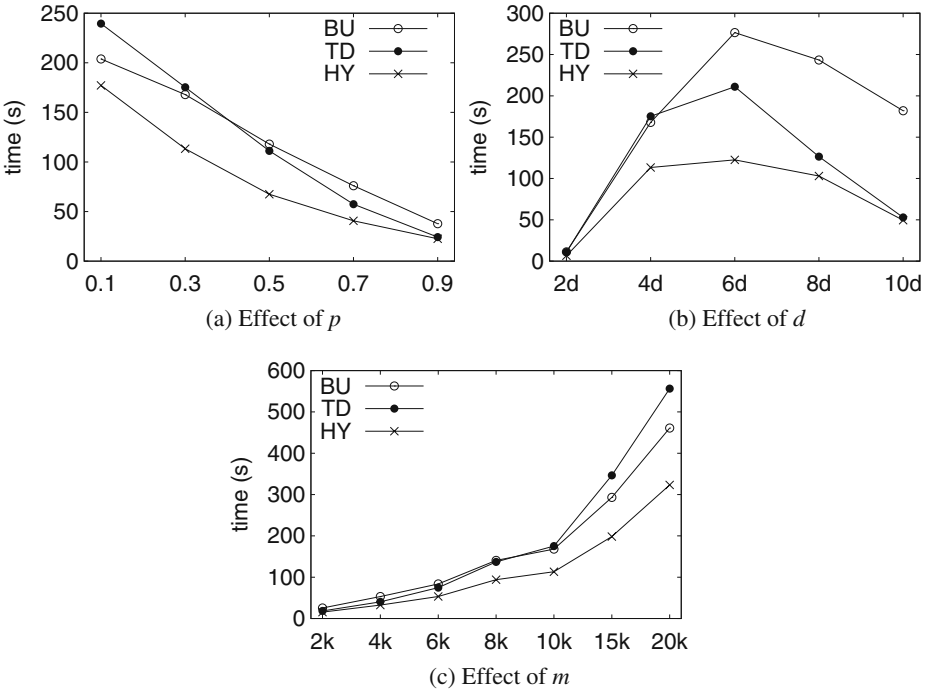


Fig. 18 Scalability with respect to probability threshold

algorithm outperforms BU and TD and reduces the worst case runtime by 50% (e.g., case 6d in Fig. 18b and case 20k in Fig. 18c).

Figure 18a shows that the runtime of the three algorithms decreases as the probability threshold p increases from 0.1 to 0.9, because when p becomes larger, there are less p-skyline objects and it is easier to prune non-skyline objects. We can also see that BU performs better than TD when $p < 0.5$, while worse when $p \geq 0.5$. As described in Section 6, BU needs to process at least $p \times 100\%$ instances of a skyline object to boost the lower bound to at least p. Thus, BU runs slower than TD when p is large.

In Fig. 18b, the runtime of the three algorithms increases when the dimensionality increases from 2 to 6, but decreases afterward. On the one hand, the cost of dominance testing between two instances, which is the basic operation in both algorithms, increases as the dimensionality increases. On the other hand, the average number of possible dominating objects for an uncertain object decreases since the data set becomes sparser when the dimensionality increases. The trend of runtime reflects the compromise of the two factors.

The higher the dimensionality, the sparser the data set. The larger the cardinality, the denser the data set. Figure 18b and c indicate that TD performs better when the data set is sparser. In sparse data sets, the subset instances of uncertain objects may have a smaller chance to overlap, and a better chance to be pruned by some subset instances of other objects. Besides, objects in sparse data sets are likely to have large skyline probabilities, since for each object, there are less possible dominating objects

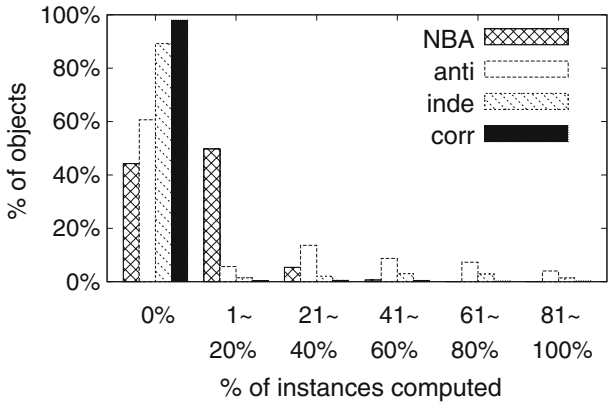


Fig. 19 Pruning effect in BU

and the dominance relations are weaker. This is not a good case for BU as it takes high cost for BU to accumulate lower bounds. Thus, TD has better performance. On the other hand, in dense data sets, the skyline probability of an instance may improve the bounds of the probabilities of more other instances and objects, and BU performs better.

8.2.3 Effectiveness of pruning techniques

The performance of BU mainly depends on the efficiency of pruning instances and objects so that their skyline probabilities do not need to be computed. Figure 19 counts, for each object, the percentage of instances whose skyline probabilities are computed by BU. We group the objects by the percentage in six ranges, and count the proportion of each group in the whole data set. It is clear that more than 90% of the objects in the NBA, independent, and correlated data sets are pruned after 20% of the instances are processed. Even for anti-correlated data sets, the corresponding figure is 66%. The pruning is more effective on independent and correlated data sets. That explains the difference of runtime on synthetic data sets.

Figure 20 counts the percentage of objects pruned by the pruning rules in BU (Section 4.2). Rule 3 is not counted since it prunes instances only. Every rule

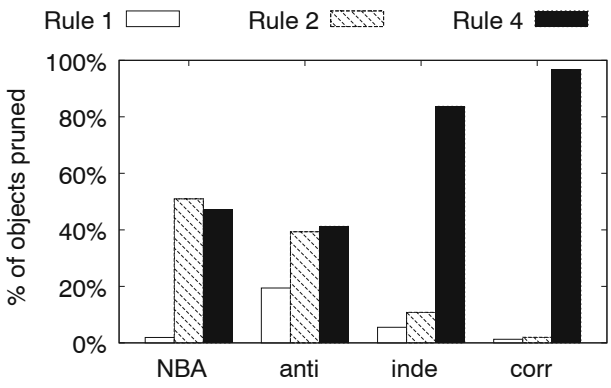
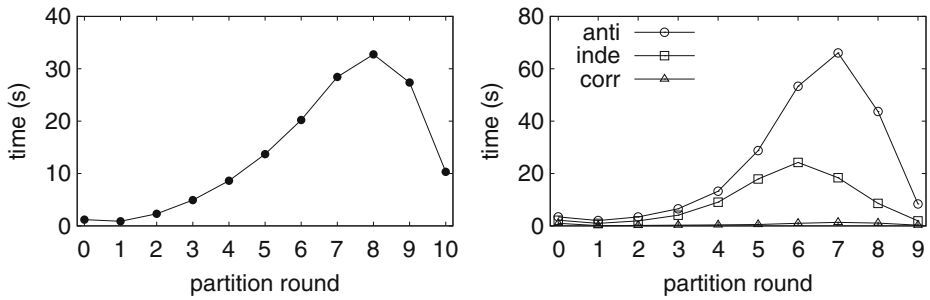


Fig. 20 Effectiveness of pruning in BU



(a) NBA data set (b) Synthetic data sets

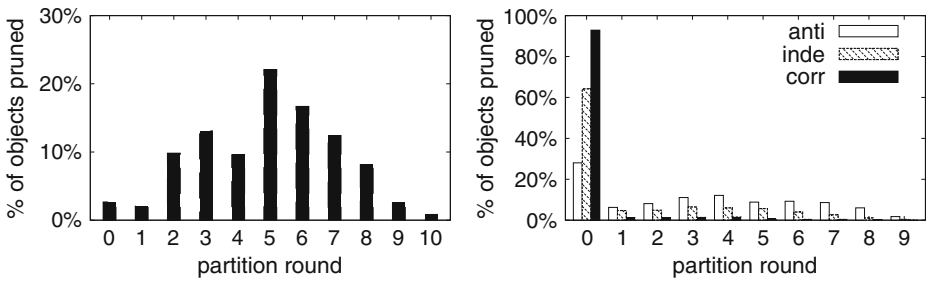
Fig. 21 Pruning effect in TD—runtime in each round

takes effect in some situations. Rule 4 is particularly effective on independent and correlated data sets where 84 and 97% objects are pruned, respectively. In those data sets, it is more likely that an object is completely dominated by another.

Figures 21 and 22 examine the effectiveness of the pruning techniques in TD. Figure 21a and b show the runtime of each round of partitioning on the NBA data set and the synthetic data sets, respectively. On all data sets, the runtime increases at first, since after such a round the leaf nodes not pruned are partitioned into more nodes. The runtime decreases in the later rounds. This is because the effectiveness of pruning in TD becomes stronger when the leaf nodes are smaller, such that the numbers of remaining objects and nodes decrease significantly.

Figure 22 shows in each round the number of objects whose probabilistic skyline memberships are determined. Recall that in round 0, the top-down method prunes uncertain objects using their MBBs. On the NBA data set, rounds 2–8 prune most of the objects, while on the synthetic data sets, most of the objects are pruned in the first round. Again, the pruning is more effective on independent and correlated data sets.

In summary, our two algorithms are effective and efficient in computing probabilistic skylines. They are also scalable on our large data sets containing millions of instances.



(a) NBA data set (b) Synthetic data sets

Fig. 22 Pruning effect in TD—percentage of objects pruned in each round

9 Discussion and conclusions

In this paper, we extended the well-known skyline analysis to uncertain data, and developed four efficacious algorithms to tackle the problem of computing probabilistic skylines on uncertain data. Using real data sets and synthetic data sets, we illustrated the effectiveness of probabilistic skylines and the efficiency and scalability of our algorithms.

Although we focused on the discrete case, some of our ideas can be applied to handle the continuous case, i.e., each uncertain object is represented by a probability density function. For example, in the top-down algorithm, for each uncertain object, we can initially partition the space into two regions such that the probability of the object in each region is 0.5. Each region can be represented by a bounding box. We can estimate the skyline probabilities of the bounding boxes and recursively partition the bounding boxes into smaller ones until the skyline probabilities of uncertain objects can be determined against the threshold. A detailed exploration on efficient methods for computing probabilistic skylines on continuous data will be given by another study.

Advanced data analysis on uncertain data is an interesting direction. In the future, we plan to exploit the probabilistic skyline analysis in real applications, and explore more analytical tasks on uncertain data.

References

Abiteboul, S., Kanellakis, P., & Grahne, G. (1987). On the representation and querying of sets of possible worlds. In *Proceedings of the 1987 ACM SIGMOD international conference on Management of data (SIGMOD'87)* (pp. 34–48). New York: ACM Press.

Aggarwal, C. C., & Yu, P. S. (2007). *A survey of uncertain data algorithms and applications*. IBM technical report (RC 24394).

Atallah, M. J., & Qi, Y. (2009). Computing all skyline probabilities for uncertain data. In *Proceedings of the twenty-eigth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems, PODS* (pp. 279–287).

Balke, W. T., Güntzer, U., & Zheng, J. X. (2004). Efficient distributed skylining for web information systems. In *EDBT 2004, 9th international conference on extending database technology* (pp. 256–273).

Benjelloun, O., Sarma, A. D., Halevy, A., & Widom, J. (2006). Uldbs: Databases with uncertainty and lineage. In *VLDB'2006: Proceedings of the 32nd international conference on very large data bases, VLDB endowment* (pp. 953–964).

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM (CACM), 18*(9), 509–517.

Bentley, J. L., Kung, H. T., Schkolnick, M., & Thompson, C. D. (1978). On the average number of maxima in a set of vectors and applications. *Journal of the ACM, 25*(4), 536–543.

Böhm, C., Fiedler, F., Oswald, A., Plant, C., & Wackersreuther, B. (2009). Probabilistic skyline queries. In *Proceedings of the 18th ACM conference on information and knowledge management, CIKM* (pp. 651–660).

Borzsonyi, S., Kossmann, D., & Stocker, K. (2001). The skyline operator. In *Proceedings of 2001 international conferences on data engineering (ICDE'01)*. Heidelberg, Germany.

Burdick, D., Deshpande, P. M., Jayram, T. S., Ramakrishnan, R., & Vaithyanathan, S. (2005). OLAP over uncertain and imprecise data. In *VLDB '05: Proceedings of the 31st international conference on very large data bases, VLDB endowment* (pp. 970–981).

Chan, C. Y., Eng, P. K., & Tan, K. L. (2005). Stratified computation of skylines with partially-ordered domains. In *Proceedings of the 2005 ACM SIGMOD international conference on management of data (SIGMOD)* (pp. 203–214).

Chan, C. Y., Jagadish, H. V., Tan, K. L., Tung, A. K. H., & Zhang, Z. (2006a). Finding *k*-dominant skylines in high dimensional space. In *Proceedings of the 2006 ACM SIGMOD international conference on management of data (SIGMOD)* (pp. 503–514).

Chan, C. Y., Jagadish, H. V., Tan, K. L., Tung, A. K. H., & Zhang, Z. (2006b). Finding *k*-dominant skylines in high dimensional space. In *SIGMOD* (pp. 503–514). New York: ACM Press.

Chan, C. Y., Jagadish, H. V., Tan, K. L., Tung, A. K. H., & Zhang, Z. (2006c). On high dimensional skylines. In *10th international conference on extending database technology (EDBT)* (pp. 478–495).

Chen, L., & Lian, X. (2008). Dynamic skyline queries in metric spaces. In *EDBT* (pp. 333–343).

Cheng, R., Kalashnikov, D. V., & Prabhakar, S. (2003). Evaluating probabilistic queries over imprecise data. In *Proceedings of the 2003 ACM SIGMOD international conference on management of data (SIGMOD'03)* (pp. 551–562). New York: ACM Press.

Cheng, R., Xia, Y., Prabhakar, S., Shah, R., & Vitter, J. S. (2004). Efficient indexing methods for probabilistic threshold queries over uncertain data. In *Proceedings of 30th international conference on very large data bases (VLDB)* (pp. 876–887).

Chomicki, J., Godfrey, P., Gryz, J., & Liang, D. (2003). Skyline with presorting. In *Proceedings of the 19th international conference on data engineering (ICDE)* (pp. 717–816).

Dai, X., Yiu, M. L., Mamoulis, N., Tao, Y., & Vaitis, M. (2005). Probabilistic spatial queries on existentially uncertain data. In *Proceeding of the 9th international symposium on spatial and temporal databases (SSTD)* (pp. 400–417).

Dalvi, N. N., & Suciu, D. (2004). Efficient query evaluation on probabilistic databases. In *Proceedings of 30th international conference on very large data bases (VLDB)* (pp. 864–875).

Dalvi, N. N., & Suciu, D. (2007). Management of probabilistic data: Foundations and challenges. In *Proceedings of the twenty-sixth ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems* (pp. 1–12). New York: ACM Press.

Dellis, E., & Seeger, B. (2007). Efficient computation of reverse skyline queries. In *Proceedings of the 33rd international conference on very large data bases (VLDB)* (pp. 291–302).

Denis Mindolin, J. C. (2009). Discovering relative importance of skyline attributes. In *Proceedings of the 35th international conference on very large data bases (VLDB)*.

Deshpande, A., & Sarawagi, S. (2007). Probabilistic graphical models and their role in databases. In *Proceedings of the 33rd international conference on very large data bases* (pp. 1435–1436).

Godfrey, P., Shipley, R., & Gryz, J. (2005). Maximal vector computation in large data sets. In *VLDB*. Trondheim, Norway.

Guttman, A. (1984). R-tree: A dynamic index structure for spatial searching. In *Proc. 1984 ACM-SIGMOD int. conf. management of data (SIGMOD'84)* (pp. 47–57). Boston, MA.

Huang, Z., Jensen, C. S., Lu, H., & Ooi, B. C. (2006). Skyline queries against mobile lightweight devices in manets. In *Proceedings of the 22nd international conference on data engineering (ICDE'06)*. New York: IEEE.

Imielinski, T., & Witold Lipski, J. (1984). Incomplete information in relational databases. *Journal of the ACM, 31*(4), 761–791.

Jiang, B., & Pei, J. (2009). Online interval skyline queries on time series. In *Proceedings of the 25th international conference on data engineering (ICDE'09)*. Shanghai, China.

Jiang, B., Pei, J., Lin, X., Cheung, D. W., & Han, J. (2008). Mining preferences from superior and inferior examples. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 390–398). New York: ACM Press.

Kossmann, D., Ramsak, F., & Rost, S. (2002). Shooting stars in the sky: An online algorithm for skyline queries. In *Proc. 2002 int. conf. on very large data bases (VLDB'02)*. Hong Kong, China.

Kriegel, H. P., Kunath, P., Pfeifle, M., & Renz, M. (2006). Probabilistic similarity join on uncertain data. In *Proceeding of the 11th international conference on database systems for advanced applications (DASFAA)* (pp. 295–309).

Kung, H. T., Luccio, F., & Preparata, F. P. (1975). On finding the maxima of a set of vectors. *Journal of the ACM, 22*(4), 469–476.

Lian, X., & Chen, L. (2008). Monochromatic and bichromatic reverse skyline search over uncertain databases. In *SIGMOD conference* (pp. 213–226).

Lin, X., Yuan, Y., Wang, W., & Lu, H. (2005). Stabbing the sky: Efficient skyline computation over sliding windows. In *Proceedings of the 21st international conference on data engineering (ICDE)* (pp. 502–513).

Morse, M. D., Patel, J. M., & Grosky, W. I. (2006). Efficient continuous skyline computation. In *Proceedings of the 22nd international conference on data engineering (ICDE)* (p. 108).

Papadias, D., Tao, Y., Fu, G., & Seeger, B. (2003). An optimal and progressive algorithm for skyline queries. In *Proceedings of the 2003 ACM SIGMOD international conference on management of data (SIGMOD)* (pp. 467–478).

Park, S., Kim, T., Park, J., Kim, J., & Im, H. (2009). Parallel skyline computation on multicore architectures. In *Proceedings of the 25th international conference on data engineering, ICDE* (pp. 760–771).

Pei, J., Jin, W., Ester, M., & Tao, Y. (2005). Catching the best views in skyline: A semantic approach. In *Proceedings of the 31st international conference on very large data bases (VLDB'05)*.

Pei, J., Fu, A. W. C., Lin, X., & Wang, H. (2007a). Computing compressed skyline cubes efficiently. In *Proceedings of the 23rd international conference on data engineering (ICDE'07)*. IEEE, Istanbul.

Pei, J., Jiang, B., Lin, X., & Yuan, Y. (2007b). Probabilistic skylines on uncertain data. In *Proceedings of the 33rd international conference on very large data bases (VLDB'07)*. Viena, Austria.

Sacharidis, D., Papadopoulos, S., & Papadias, D. (2009). Topologically sorted skylines for partially ordered domains. In *Proceedings of the 25th international conference on data engineering, ICDE* (pp. 1072–1083).

Sarma, A. D., Benjelloun, O., Halevy, A. Y., & Widom, J. (2006). Working models for uncertain data. In *Proceedings of the 22nd international conference on data engineering (ICDE)* (p. 7).

Sarma, A. D., Lall, A., Nanongkai, D., & Xu, J. (2009). Randomized multi-pass streaming skyline algorithms. In *Proceedings of the 35th international conference on very large data bases*.

Sen, P., Deshpande, A., & Getoor, L. (2007). Representing tuple and attribute uncertainty in probabilistic databases. In *Workshops proceedings of the 7th IEEE international conference on data mining (ICDM)* (pp. 507–512). Los Alamitos: IEEE Computer Society.

Sharifzadeh, M., & Shahabi, C. (2006). The spatial skyline queries. In *Proceedings of the 32nd international conference on very large data bases (VLDB)* (pp. 751–762).

Soliman, M. A., Ilyas, I. F., & Chang, K. C. C. (2007). Top-*k* query processing in uncertain databases. In *Proceedings of the 23rd international conference on data engineering (ICDE'07)*. New York: IEEE.

Tan, K. L., Eng, P. K., & Ooi, B. C. (2001). Efficient progressive skyline computation. In *Proceedings of 27th international conference on very large data bases (VLDB)* (pp. 301–310).

Tao, Y., & Papadias, D. (2006). Maintaining sliding window skylines on data streams. *IEEE Transactions on Knowledge and Data Engineering, 18*(2), 377–391.

Tao, Y., Cheng, R., Xiao, X., Ngai, W. K., Kao, B., & Prabhakar, S. (2005). Indexing multidimensional uncertain data with arbitrary probability density functions. In *Proceedings of 31st international conference on very large data bases (VLDB)* (pp. 922–933).

Tao, Y., Xiao, X., & Pei, J. (2006). Subsky: Efficient computation of skylines in subspaces. In *Proceedings of the 22nd international conference on data engineering (ICDE'06)*. New York: IEEE.

Wong, R. C. W., Pei, J., Fu, A. W. C., & Wang, K. (2007). Mining favorable facets. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 804–813). New York: ACM.

Wu, P., Zhang, C., Feng, Y., Zhao, B. Y., Agrawal, D., & Abbadi, A. E. (2006). Parallelizing skyline queries for scalable distribution. In *Proceedings of the 10th international conference on extending database technology (EDBT'06)*. Munich: Springer.

Xia, T., & Zhang, D. (2006). Refreshing the sky: The compressed skycube with efficient support for frequent updates. In *Proceedings of the 2006 ACM SIGMOD international conference on management of data (SIGMOD'06)* (pp. 491–502). New York: ACM Press.

Yuan, Y., Lin, X., Liu, Q., Wang, W., Yu, J. X., & Zhang, Q. (2005). Efficient computation of the skyline cube. In *Proceedings of the 31st international conference on very large data bases (VLDB)* (pp. 241–252).

Zhang, W., Lin, X., Zhang, Y., Wang, W., & Yu, J. X. (2009a). Probabilistic skyline operator over sliding windows. In *Proceedings of the 25th international conference on data engineering, ICDE* (pp. 1060–1071).

Zhang, Z., Cheng, R., Papadias, D., & Tung, A. K. H. (2009b). Minimizing the communication cost for continuous skyline maintenance. In *Proceedings of the ACM SIGMOD international conference on management of data*. Providence, RI, USA.

Zhang, Z., Yang, Y., Cai, R., Papadias, D., & Tung, A. K. H. (2009c). Kernel-based skyline cardinality estimation. In *Proceedings of the ACM SIGMOD international conference on management of data*.