Skyline Probability over Uncertain Preferences

Qing Zhang[*‡], Pengjie Ye[*‡]
[*]The Australian e-Health Research Centre
{qing.zhang, pengjie.ye}@csiro.au

Xuemin Lin[‡], Ying Zhang[‡]
[‡]The University of New South Wales, Australia
{lxue, yingz}@cse.unsw.edu.au

ABSTRACT

Skyline analysis is a key in a wide spectrum of real applications involving multi-criteria optimal decision making. In recent years, a considerable amount of research has been contributed on efficient computation of skyline probabilities over uncertain environment. Most studies if not all, assume uncertainty lies only in attribute values. To the extent of our knowledge, only one study addresses the skyline probability computation problem in scenarios where uncertainty resides in attribute preferences, instead of values. However this study takes a problematic approach by assuming *independent object dominance*, which we find is not always true in uncertain preference scenarios. In fact this assumption has already been shown to be not necessarily true in uncertain value scenarios. Motivated by this, we revisit the skyline probability computation over uncertain preferences in this paper.

We first show that the problem of skyline probability computation over uncertain preferences is ♯P-complete. Then we propose efficient exact and approximate algorithms to tackle this problem. While the exact algorithm remains exponential in the worst case, our experiments demonstrate its efficiency in practice. The approximate algorithm achieves ϵ-approximation by the confidence $(1-\delta)$ with time complexity $O(dn\frac{1}{\epsilon^2}\ln\frac{1}{\delta})$, where n is the number of objects and d is the dimensionality. The efficiency and effectiveness of our methods are verified by extensive experimental results on real and synthetic data sets.

1. INTRODUCTION

Given two multi-dimensional objects p and q, p dominates q *iff* p is preferred to q in all dimensions, and strictly better than q in at least one dimension. A skyline point is a point that are not dominated by any other points in the same data set. Skyline computation aims at efficiently retrieving all skyline points from a (possibly) large data set [4]. Since the debut of the skyline query processing research in 2001, this topic has been extensively studied. Various techniques have been developed to compute skyline as well as its variations such as reverse skyline, skycube, etc. In 2007, driven by many applications involving uncertain data, Pei et. al first investigated efficient skyline computation on uncertain data [19], where a multi-dimensional object is assumed to have uncertain attribute values under certain probability distributions. Obviously under this assumption, any object can be a skyline point with certain probability $p \in [0,1]$, where p is defined as the skyline probability. The traditional skyline definition is thus naturally extended to probabilistic skyline which consists of all points whose skyline probabilities are over a threshold τ $(0 < \tau < 1)$. Many techniques have been developed thereafter to efficiently compute the probabilistic skyline over uncertain data. Readers can check the related work section for details.

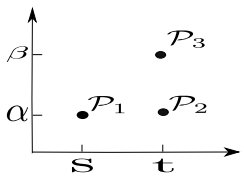
Most existing skyline probability computation methods assume that computations are performed in a database with deterministic attribute preferences. However in many applications, user preferences towards particular data values are different, especially when categorical data is involved. For example, a music fan prefers Mozart's brisk minuet while another may like Beethoven's pastoral symphony. Even a same user may change his/her own preferences on two attributes under different situations, such as a tourist favouring a beach view room in scorching summer and preferring a fireplace room in chilly winter. To the extent of our knowledge, the only study on skyline probability computation over uncertain preferences was published in literature in 2010, where Sacharidis et al. first investigated computing probabilistic contextual skylines [21]. In that study, every multi-dimensional object has fixed attribute values while preferences among those values are uncertain. Mathematically, they model the preference π between two values as a random variable such that $0 \leq \pi \leq 1$, where $\pi = 0/1$ degenerates π to the traditional certain preference definition. Note that this probabilistic preference model has already been widely used in voting theory as fuzzy/probability voting schema [8] and probabilistic majority rules [17].

To compute the skyline probability on uncertain preferences, Sacharidis et al. [21] adopted the *independent object dominance* assumption. In a nutshell, this assumption treats object dominances as mutually independent events. Therefore the skyline probability of an object can be simply calculated as the multiplication of probabilities that this object is not dominated by others, mathematically represented as Equation 2 in [21]. As we already knew that this assumption is not necessarily true in skyline probability computation over uncertain values [19]. Here we also demonstrate

EDBT/ICDT '13 March 18 - 22 2013, Genoa, Italy

that in general, this assumption is also **NOT** valid when considering skyline probability over uncertain preferences, through the following observation:

Observation. Consider a two-dimensional space with three objects in Figure 1 (left). We use $\prec$ to represent preference/dominance relations between values/objects. For example, $s \prec t$ implies that s is preferred to t by the population. Also we assume that any two attribute values are equally preferred by the population, as shown in Figure 1 (right).



$\Pr(s \prec t) = \frac{1}{2}$ $\Pr(t \prec s) = \frac{1}{2}$

$\Pr(\alpha \prec \beta) = \frac{1}{2}$ $\Pr(\beta \prec \alpha) = \frac{1}{2}$

Figure 1: A 2-D space with uncertain preferences

Without loss of generality, we consider the skyline probability of $\mathcal{P}_1$. It is easy to see that the probability of $\mathcal{P}_2$ dominating $\mathcal{P}_1$ is $\Pr(\mathcal{P}_2 \prec \mathcal{P}_1) = \frac{1}{2}$. Similarly we have $\Pr(\mathcal{P}_3 \prec \mathcal{P}_1) = \frac{1}{4}$. Following approaches taken in [21], shortened as *Sac* hereafter, by assuming independent object dominance we compute the skyline probabilities of $\mathcal{P}_1$ as:

$$sky(\mathcal{P}_1) = (1 - \frac{1}{2}) * (1 - \frac{1}{4}) = \frac{3}{8}.$$

Alternatively, following basic definitions of probability, we always can take a *naive* approach to compute skyline probabilities, i.e. enumerating all sample spaces and summing probabilities where $\mathcal{P}_1$ is a skyline point. In Figure 2 we list all possible sample spaces formed by various combinations of preferences, with corresponding probability and skyline objects. Hence to compute $sky(\mathcal{P}_1)$, we only need to add up

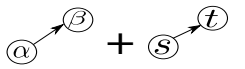
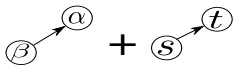
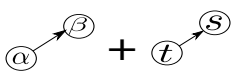
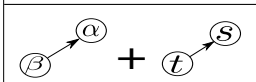
Sample space	Probability	Skyline objects
	$\frac{1}{4}$	$\mathcal{P}_1$
	$\frac{1}{4}$	$\mathcal{P}_1, \mathcal{P}_3$
	$\frac{1}{4}$	$\mathcal{P}_2$
	$\frac{1}{4}$	$\mathcal{P}_3$

Figure 2: All sample spaces with skyline points

probabilities of spaces $(\alpha \prec \beta \wedge s \prec t)$ and $(\beta \prec \alpha \wedge s \prec t)$:

$$sky(\mathcal{P}_1) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}.$$

Note that the probabilities of sample spaces are computed by assuming independence among attribute preferences of different dimensions.

Conclusion. The second naïve approach enumerates all possible sample spaces and outputs correct results. Therefore the $sky(\mathcal{P}_1)$ computed by Sac is not correct. In fact for three objects in this example Sac can only correctly compute $sky(\mathcal{P}_2)$.

Analyses. So why Sac fails on $sky(\mathcal{P}_1)$ and $sky(\mathcal{P}_3)$? The answer is simple: with uncertain preference, object dominance relations can not be assumed independent. Intuitively the dominance relations between two objects will be related to uncertain preferences between their attribute values. If two objects share a common attribute value, such as $\mathcal{P}_2$ and $\mathcal{P}_3$ sharing t, the dominance relations involving these two are no longer independent. Let $Dom(,)$ define dominance relations between two objects. Then in our observation, $Dom(\mathcal{P}_1, \mathcal{P}_2)$, $Dom(\mathcal{P}_1, \mathcal{P}_3)$ are not mutually independent. On the other hand when no common values exist between two objects, dominance relations involving these two can be considered independent. This explains why Sac can correctly compute $sky(\mathcal{P}_2)$ since $\mathcal{P}_1$ and $\mathcal{P}_3$ share no values, and $Dom(\mathcal{P}_1, \mathcal{P}_2)$ and $Dom(\mathcal{P}_3, \mathcal{P}_2)$ are thus mutually independent.

Indeed, our later theoretic analyses indicate that within uncertain preference scenarios, computing only a single object's skyline probability becomes elusive, not even to say finding all objects' skyline probabilities, i.e. probabilistic skyline. In this paper, we thus focus on efficiently computing the skyline probability of a given object. Our main contributions are:

- We propose a deterministic algorithm and prove that skyline probability computation over uncertain preferences is a ♯P-complete problem.
- We design a randomised algorithm to compute skyline probabilities in large data sets. Our algorithm achieves ϵ-approximation with confidence $(1 - \delta)$, within time complexity $O(dn\frac{1}{\epsilon^2}\ln\frac{1}{\delta})$, where d is the dimensionality and n is the cardinality of the data sets.
- We propose two speed-up techniques that can efficiently improve performances of both algorithms.
- We perform extensive experiments on real and large synthetic data sets. The performance of our algorithms is verified by our experiment results.

The rest of the paper is organised as follows. Section 2 introduces notations and definitions used throughout this paper and also formally defines the problem of skyline probability computation over uncertain preferences. Section 3 proposes a deterministic solution with ♯P-complete proofs. Section 4 presents an ϵ-approximate approach on computing the skyline probability. Section 5 discusses two speed-up techniques aiming at improving computational efficiencies of deterministic and approximate approaches. Section 6 reports a systematic performance study on real and synthetic data sets. Section 7 reviews related works in the literature and Section 8 concludes this paper with possible future works.

2. UNCERTAIN PREFERENCES AND SKYLINE PROBABILITY

Given a multi-dimensional space D ($|D| = d$) consisting of $n+1$ objects: $\mathcal{O}$ and $\mathcal{Q}_i$ ($1 \leq i \leq n$). Without loss of generality, we consider calculating the skyline probability of $\mathcal{O}$. For any object $\mathcal{O}$ ($\mathcal{Q}_i$), we use $\mathcal{O}.j$ ($\mathcal{Q}_i.j$) to represent its value on jth dimension. Also we use $\prec$ to denote preference/dominance relations among values/objects. Figure 3 summarises the above notations and others used throughout this paper.

Notation	*Name*
D	d-dimensional space
j	jth dimension of D, $1 \leq j \leq d$
$\mathcal{O}$, $\mathcal{Q}_i$ ($1 \leq i \leq n$)	$n+1$ distinct points in D
$\mathcal{O}.j$ ($\mathcal{Q}_i.j$)	the jth dimensional value of $\mathcal{O}$ ($\mathcal{Q}_i$)
$\prec$	preference / dominance relations
e_i	object dominance event $\mathcal{Q}_i \prec \mathcal{O}$
E_I	intersection of events $\{e_i\}, i \in I$
$sky(\mathcal{O})$	$\mathcal{O}$'s skyline probability

Figure 3: The summary of notations

Uncertain preference. Given two distinct values α and β, we use probabilistic models to describe the uncertain preferences between them. That is:

$$\Pr(\alpha \prec \beta) + \Pr(\beta \prec \alpha) \leq 1.$$

Here the inequality stands for chances when α and β are incomparable. If α and β are the same value, we have $\Pr(\alpha \preceq \beta) = \Pr(\beta \preceq \alpha) = 1$.

Dominance probability. For reasons of simplicity, we assume no duplicate objects in D. Given two objects $\mathcal{Q}_i$ and $\mathcal{O}$, $\mathcal{Q}_i$ dominates $\mathcal{O}$ (i.e. $\mathcal{Q}_i \prec \mathcal{O}$) *iff* $\mathcal{Q}_i$ is preferred or equal to $\mathcal{O}$ on any dimension and preferred on at least one dimension (i.e. $\forall j\ \mathcal{Q}_i.j \preceq \mathcal{O}.j \wedge \exists j'\ \mathcal{Q}_i.j' \prec \mathcal{O}.j'$). Let e_i denote the event $\mathcal{Q}_i \prec \mathcal{O}$, by our definition, the probability of e_i is a joint probability of attribute value preferences:

$$\Pr(e_i) = \Pr(\bigcap_{j=1}^{d}(\mathcal{Q}_i.j \preceq \mathcal{O}.j)) \quad (1)$$

Equation 1 can be simplified through assuming *attribute value preferences of different dimensions are mutually independent*, which is a commonly adopted assumption in multi-dimensional data analyses. Therefore, we have:

$$\Pr(e_i) = \prod_{j=1}^{d} \Pr(\mathcal{Q}_i.j \preceq \mathcal{O}.j) \quad (2)$$

Skyline Probability. An object's skyline probability is defined as the probability that this point can **not** be dominated by others. Therefore the skyline probability of $\mathcal{O}$, $sky(\mathcal{O})$, can be mathematically represented as:

$$sky(\mathcal{O}) = \Pr(\bigcap_{i=1}^{n} \overline{e_i}) = 1 - \Pr(\bigcup_{i=1}^{n} e_i) \quad (3)$$

Note here $\overline{e_i}$ denotes the complementary event of e_i. From the *inclusion-exclusion principle* [15], we rewrite Equation 3 as:

$$sky(\mathcal{O}) = 1 + \sum_{k=1}^{n}(-1)^k \sum_{\substack{I \subset \{1,\ldots,n\} \\ |I|=k}} \Pr(E_I), \quad (4)$$

where E_I denotes the intersection of $|I|$ events, i.e. $\bigcap_{i \in I} e_i$.

As discussed, those I object dominance events in E_I are generally not mutually independent, i.e. $\Pr(E_I) \neq \prod_{i \in I} \Pr(e_i)$. Indeed for $\mathcal{Q}_i$ involved in E_I, let $\mathbb{V}_j^I$ denote the set of *distinct values* of those $\mathcal{Q}_i.j$. We can rewrite $\Pr(E_I)$ as:

$$\Pr(E_I) = \Pr(\bigcap_{j=1}^{d} \bigcap_{v \in \mathbb{V}_j^I} v \preceq \mathcal{O}.j) \quad (5)$$

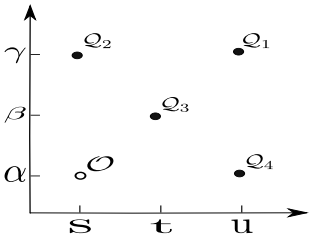
To compute Equation 5, we make another assumption: *preferences sharing a common value are mutually independent.* We illustrate this assumption through an example of three values α, β and γ with uncertain preferences predefined. We believe that preferences on (α, β) and (β, γ) are mutually independent, i.e. users' preference on (α, β), will not influence their preference on (β, γ). Note that if considering three preferences on (α, β), (β, γ) and (γ, α) together, they are usually not mutually independent since otherwise it may be against the common sense assumption that preferences should be non-conflicting. In light of this assumption, we simplify Equation 5 as:

$$\Pr(E_I) = \prod_{j=1}^{d} \prod_{v \in \mathbb{V}_j^I} \Pr(v \preceq \mathcal{O}.j), \quad (6)$$

It is easy to verify that for any $|I|$ with only one event e_i, the above equation degenerates to Equation 2.

Collectively, with Equations 4 and 6, we can compute the skyline probability of any object in a space predefined with uncertain preferences. We use the following running example to illustrate how to compute skyline probabilities from those two equations.

Example 1. Assume a two-dimensional space D including five points, shown in Figure 4 (a). All attribute values are equally preferred with probability 0.5. Figure 4 (b) lists all uncertain preferences related to $\mathcal{O}$.



(a) An example space

$\Pr(s \prec t) = \Pr(t \prec s) = 0.5$

$\Pr(s \prec u) = \Pr(u \prec s) = 0.5$

$\Pr(\alpha \prec \beta) = \Pr(\beta \prec \alpha) = 0.5$

$\Pr(\alpha \prec \gamma) = \Pr(\gamma \prec \alpha) = 0.5$

(b) uncertain preferences

Figure 4: A running example

To compute $sky(\mathcal{O})$, we first calculate all joint probabilities, $\Pr(E_I)$, in Equation 4. This can be easily achieved by applying Equation 6. For instance:

$$\Pr(e_1 \cap e_2 \cap e_3) = (\frac{1}{2})^2 \times (\frac{1}{2})^2 = \frac{1}{16}.$$

Then from Equation 4 we derive:

$$sky(\mathcal{O}) = 1 - \frac{3}{2} + \frac{17}{16} - \frac{7}{16} + \frac{1}{16} = \frac{3}{16}.$$

Again it can be verified that if assuming object dominance independent, we will have an incorrect result of $sky(\mathcal{O})$, $\frac{9}{64}$.

Problem Statement: Given a set of objects in a d-dimensional space D with uncertain preferences predefined, in this paper we investigate the problem of efficiently computing the skyline probability of a given object $\mathcal{O}$ in this set.

3. DETERMINISTIC ALGORITHM

An immediate deterministic algorithm, with applying Equations 4 and 6, can be easily derived. However instead of naïvely calculating every $\Pr(E_I)$ with time complexity $O(d\,|I|)$, we can slightly speed up these calculations through a simple sharing computation technique. That is by organising those $\Pr(E_I)$ calculations systematically, we can achieve a constant time complexity, $O(d)$, on computing $\Pr(E_I)$.

Consider two sets of events, I and $I-\{e_i\}$. From Equation 6, we have:

$$\Pr(E_I) = \Pr(E_{I-\{e_i\}}) * \prod_{j'} \Pr(\mathcal{Q}_i.j' \prec \mathcal{O}.j')$$

Here j' is a dimension where $\mathcal{Q}_i.j' \notin \mathbb{V}_{j'}^{I-\{e_i\}}$, i.e. $\mathcal{Q}_i.j'$ does not equal to any j'-dimensional values of points involved in $I - \{e_i\}$. Given the value of $\Pr(E_{I-\{e_i\}})$, since there exist at most d such dimensions, the computational complexity for any $\Pr(E_I)$ would be $O(d)$, For instance, in our running example, if given $\Pr(e_1 \cap e_2) = \frac{1}{4}$, we can compute $\Pr(e_1 \cap e_2 \cap e_3)$ as easy as:

$$\Pr(e_1 \cap e_2 \cap e_3) = \Pr(e_1 \cap e_2) * \frac{1}{2} * \frac{1}{2} = \frac{1}{16}.$$

To apply this sharing technique, we need to organise all $\Pr(E_I)$ computations systematically. A rule of thumb is to always compute $\Pr(E_I)$ from $\Pr(E_{I'})$, where $|I| = |I'| + 1$. Figure 5 illustrates a possible computational sequence, indicated by arrows, in finding $sky(\mathcal{O})$ in our running example. It is easy to see that this rule guarantees that any $\Pr(E_I)$

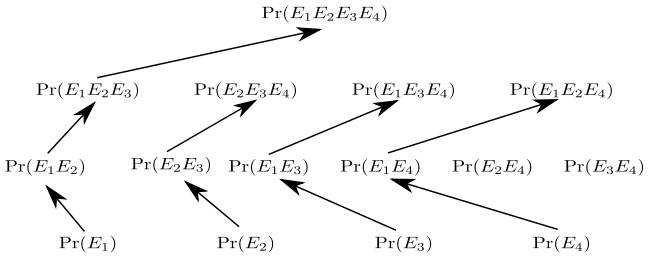


Figure 5: A sharing computation sequence for $sky(\mathcal{O})$ in Example 1

be computed in $O(d)$. Algorithm 1 concludes this basic deterministic algorithm from above analyses.

Time complexity. In Algorithm 1, we compute every $\Pr(E_I)$ in $O(d)$. Therefore the time complexity of this algorithm is:

$$O(\sum_{k=1}^{n} d \cdot \binom{n}{k}) = O(d2^n).$$

Algorithm 1: A deterministic algorithm

Input: d-dimensional points $\mathcal{O}, \mathcal{Q}_1, ..., \mathcal{Q}_n$ in space D with uncertain preferences already defined on D
Output: $sky(\mathcal{O})$

```
Y ← 1;
for k=1 to n do
    if k equals 1 then
        compute all n probabilities Pr(E_I);
    else
        compute all (n choose k) joint probabilities Pr(E_I) where
        |I| = k, from already computed (n choose k-1)
        probabilities Pr(E_I') where |I'| = k − 1;
    Y ← Y + (−1)^k · Σ Pr(E_I);
Return Y;
```

Obviously when n becomes large, the cost of this deterministic solution will eventually be prohibitive. Can we do better? Unfortunately we prove that there unlikely exists any deterministic approach with polynomial time complexity, as stated in the following theorem.

3.1 Computational complexity analyses

THEOREM 1. *Computing the skyline probability of an object in a multi-dimensional space with predefined uncertain preferences, is ♯P-complete.*

PROOF. We first show that this problem is in the class ♯P. Again considering $sky(\mathcal{O})$ in a space D with $n+1$ points, $(\mathcal{O}, \mathcal{Q}_1, \ldots, \mathcal{Q}_n)$. Assume all attribute values in D are comparable with unanimous uncertain preferences $\frac{1}{2}$. Given a set of preference assignments, we define it a successful one if $\mathcal{O}$ is a skyline point under these assignments. In this space D, any set of preference assignments occurs with a same probability: $\frac{1}{2}^{-d+\sum_{j=1}^{d} \mathbb{V}_j}$, here $\mathbb{V}_j$ represents number of distinct values in jth dimension. Therefore computing $sky(\mathcal{O})$ is equivalent to counting number of successful sets, i.e. preference assignments. A nondeterministic Turing machine needs only guess an assignment and check in polynomial time whether $\mathcal{O}$ is a skyline point under this particular assignment. Therefore computing $sky(\mathcal{O})$ is ♯P.

Then we prove it is ♯P-complete by polynomial time reducing a known ♯P-complete problem, the DNF counting [16]. Restricted to only positive literals, we have positive DNF formula, as illustrated in the following example with 4 literals and 3 clauses:

$$(x_1 \wedge x_3) \vee (x_2 \wedge x_4) \vee (x_3 \wedge x_4) \tag{7}$$

Given an instance of the positive DNF formula with d literals and n clauses, we reduce it to a case of computing $sky(\mathcal{O})$. That is from each clause C_i in the DNF formula, we define attribute values of $\mathcal{Q}_i$ as:

- if $x_j \in C_i$, $\mathcal{Q}_i.j \neq \mathcal{O}.j$;
- if $x_j \notin C_i$, $\mathcal{Q}_i.j = \mathcal{O}.j$;

Also if an x_j of C_i is true, we consider $\mathcal{Q}_i.j \prec \mathcal{O}.j$ with probability $\frac{1}{2}$. Otherwise $\mathcal{O}.j \prec \mathcal{Q}_i.j$ with probability $\frac{1}{2}$. It is easy to verify that this reduction can be finished in polynomial time.

On one hand, if there exists a satisfiable assignment of the DNF formula, i.e. at least one C_i is true, then we have $\mathcal{Q}_i \prec \mathcal{O}$. Hence the number of satisfiable assignments equals the number of preference assignments where $\mathcal{O}$ is dominated by others. Let μ represent the constant probability of any preference assignments. Assume that the solution of a DNF counting problem is U. We have:

$$sky(\mathcal{O}) = 1 - \mu * U$$

On the other hand, if $\mathcal{O}$ is dominated by at least one point in D, then the corresponding C_i must be true. This implies a satisfiable assignment of the DNF formula. Consequently, the solution U of this DNF counting problem is:

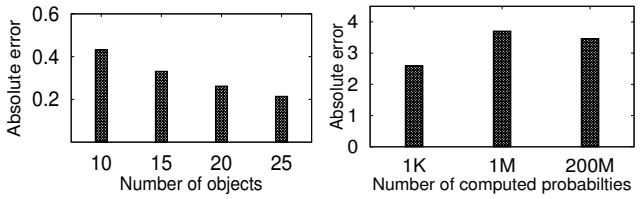
$$U = (1 - sky(\mathcal{O}))/\mu$$

□

Note that if $d = 1$, since all objects have different attribute values, the skyline probability can be easily computed in $O(n)$ by assuming independent object dominance. However Theorem 1 remains ♯P-complete whenever $d \geq 2$. Therefore in general cases the performance of any deterministic approach is limited, especially when n is large. In light of this, we propose the following approximate algorithm as a remedy solution for efficiently computing skyline probabilities of objects in a very large data set.

4. MONTE CARLO ESTIMATION

An immediate approximate solution for $sky(\mathcal{O})$ is to only consider some 'important' objects, or compute some significant joint probabilities of those $2^n - 1$ ones in Equation 4 and ignore remaining items. We did a simple trial over a synthetic data set with 1000 uniformly distributed 5-d objects to test the feasibility of these two approaches, namely $\mathcal{A}1$: computing $sky(\mathcal{O})$ from only partial number of objects who dominate $\mathcal{O}$ with highest probabilities; $\mathcal{A}2$: computing $sky(\mathcal{O})$ through only calculating partial number of joint probabilities in Equation 4. Figure 6 presents absolute errors of our two approximate approaches. Clearly, $\mathcal{A}_2$ is not a



(a) Solution $\mathcal{A}1$ (b) Solution $\mathcal{A}2$

Figure 6: Two tentative approximate solution

good approach since even a random guess will guarantee better absolute errors (less than 1). $\mathcal{A}_1$ is also not a satisfactory approach since it can not guarantee the quality of approximate answers. Moreover, it takes more than 1 hour for $\mathcal{A}_1$ to reach the best approximation result after computing 25 important objects. Longer computational time, if considering more important objects, can be envisaged. Therefore in the following subsections, we propose a sampling based Monte Carlo estimation method which not only guarantees ϵ-approximation, but also can be computed within a linear time complexity.

4.1 The sampling method

As discussed in the introduction part, we can always naïvely compute skyline probabilities from summing probabilities of sample spaces where $\mathcal{O}$ is a skyline point. Mathematically, let Ω represent the set of sample spaces, i.e. various combinations of attribute value preferences involving $\mathcal{O}$ and other objects. Let ω_h be one sample space s.t. $\omega_h \in \Omega$, $1 \leq h \leq |\Omega|$. We can express our naïve method mathematically as:

$$sky(\mathcal{O}) = \sum_{h=1}^{|\Omega|} (\Pr(\omega_h) \cdot |\{\omega_h \in \Omega | S_{\mathcal{O}}\}|) \quad (8)$$

Here $S_{\mathcal{O}}$ represents the state where $\mathcal{O}$ is a skyline point in a sample space. Recall our independent assumptions discussed in Section 2, $\Pr(\omega_h)$ can be efficiently computed through multiplications of all preference probabilities. Figure 7 lists all possible sample spaces of our running example (Figure 4), where $\mathcal{O}$ is a skyline point, together with corresponding probabilities.

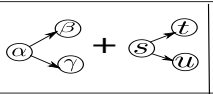
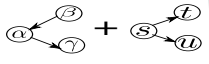
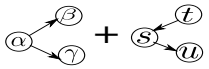
Sample space	Probability	Skyline objects
	$\frac{1}{16}$	$\mathcal{O}$
	$\frac{1}{16}$	$\mathcal{O}$, $\mathcal{Q}_3$
	$\frac{1}{16}$	$\mathcal{O}$, $\mathcal{Q}_3$

Figure 7: Sample spaces where $\mathcal{O}$ is a skyline object

It is easy to infer that $|\Omega|$ increases exponentially with the total number of distinct attribute values on all dimensions. Therefore when $|\Omega|$ becomes very large, enumerating all possible worlds becomes prohibitive. Instead we sample only part of them. Specifically, we choose a sample space ω_h with probability $\Pr(\omega_h)$. In ω_h, we check whether $\mathcal{O}$ is a skyline point. Repeat this step m times and let Y represent the number of sample spaces where $\mathcal{O}$ is a skyline point. We use $\frac{Y}{m}$ as the final estimation of $sky(\mathcal{O})$.

To sample an ω_h, we do not need to sample all its preferences. Instead, in our implementation, we sample ω_h's preferences on the fly, i.e. *lazy sampling*. With this strategy, we start by checking whether $\mathcal{O}$ is a skyline point against every other object and we only sample preferences needed in this checking so far. Whenever we observe $\mathcal{O}$ is dominated by an object, the corresponding ω_h can thus be safely discarded even we may have only partially sampled all ω_h's preferences.

Intuitively, with this lazy sampling strategy, if $\mathcal{O}$ is not a skyline object in ω_h, we expect it to be dominated by objects in our checking sequence as early as possible, if not the first. Therefore, in the very first step of our sampling algorithm, we sort all other objects according to their probabilities of dominating $\mathcal{O}$, to form a checking sequence. The object with highest probability of dominating $\mathcal{O}$ is always checked first. For instance, in our running example, we always check $\mathcal{O}$ against $\mathcal{Q}_2$ and $\mathcal{Q}_4$ first, then $\mathcal{Q}_1$ and $\mathcal{Q}_3$. Note that although this sorting incurs computational overheads, it is

almost negligible when considering all m sampling iterations can share a same sorted checking sequence.

4.2 Algorithm and analyses

Algorithm 2 collectively presents details of our sampling algorithm discussed above. The performance of this algorithm is guaranteed in the following theorem:

Algorithm 2: A Monte Carlo sampling algorithm

Input: d-dimensional objects $\mathcal{O}, \mathcal{Q}_1, \ldots, \mathcal{Q}_n$ in space D and uncertain preferences already defined on D
Output: the approximation of $sky(\mathcal{O})$

Sort $\mathcal{Q}_1, \ldots, \mathcal{Q}_n$ in descending order, basing on their dominant probabilities on $\mathcal{O}$;
$Y \leftarrow 0$;
for *h=1 to m* **do**
  for *j=1 to n* **do**
    check dominance between $\mathcal{O}$ and $\mathcal{Q}_j$ in the sorted sequence, sampling preferences accordingly ;
    if $\mathcal{O}$ *is dominated by* $\mathcal{Q}_j$ **then**
      go to 3;
  $Y \leftarrow Y + 1$;
Return Y/m;

THEOREM 2. *Given an absolute error bound ϵ and a confidence interval $1-\delta$. If $m = \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$, Algorithm 2 guarantees:*

$$\Pr\left(\left|\frac{Y}{m} - sky(\mathcal{O})\right| \geq \varepsilon\right) \leq \delta$$

PROOF. Let X_h define an independent $0-1$ random variable, which equals 1 when $\mathcal{O}$ is a skyline point under ω_h. Since every ω_h is sampled with probability $\Pr(\omega_h)$, from Equation 8 we derive that $E[X_h] = sky(\mathcal{O})$. Therefore, the expectation value of Y is:

$$E[Y] = \sum_{h=1}^{m} E[X_h] = m * \Pr(sky(\mathcal{O})) \tag{9}$$

Recall *Hoeffding's inequality* [16], we have:

$$\Pr\left(\left|\frac{Y}{m} - sky(\mathcal{O})\right| \geq \varepsilon\right) \leq 2e^{-2m\varepsilon^2}$$

Let $m = \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$, we prove this theorem. □

Time complexity. It is easy to derive that Algorithm 2 can compute an approximate skyline probability in $O(dn\frac{1}{\varepsilon^2} \ln \frac{1}{\delta})$.

To further improve this sampling algorithm's performance, we propose two speed-up preprocessing techniques in the next section aiming at reducing number of necessary objects involved in the skyline probability computation. More importantly, these techniques can also be applied on the deterministic algorithm, making exact skyline probability computation possible within polynomial time, under certain data distributions.

5. ABSORPTION AND PARTITION

In this part we introduce two speed-up preprocessing techniques, namely absorption and partition. In a nutshell, they both aim at reducing number of objects necessarily involved in skyline probability computations.

Absorption. The intuition is that to compute $sky(\mathcal{O})$ in Algorithm 1, we may not necessarily involve every other point. For instance in our running example, with/without $\mathcal{Q}_1$, we always compute same result of $sky(\mathcal{O})$. Thus $\mathcal{Q}_1$ becomes a dispensable object in computing $sky(\mathcal{O})$. We call it being absorbed. Mathematically, we can rigidly define whether an object can be absorbed by others through the following theorem:

THEOREM 3. *If a point $\mathcal{Q}_i$ shares identical values with $\mathcal{O}$ on m dimensions, $0 \leq m \leq d$. Then any point $\mathcal{Q}_j$ shares identical values with $\mathcal{Q}_i$ on all remaining $d-m$ dimensions can be 'absorbed' by $\mathcal{Q}_i$, i.e.*

$$sky(\mathcal{O}) = \Pr(\bigcap_{i=1}^{n} \overline{e_i}) = \Pr(\overline{e_1} \bigcap \ldots \bigcap \overline{e_{j-1}} \bigcap \overline{e_{j+1}} \ldots \bigcap \overline{e_n})$$

PROOF. Without loss of generality, let us assume $\mathcal{Q}_{i'}$ $(2 \leq i' \leq n)$ shares identical values with $\mathcal{O}$ on m dimensions. Also let $\mathcal{Q}_1$ shares identical values with $\mathcal{Q}_{i'}$ on remaining dimensions. Applying the *inclusion-exclusion principle*, the skyline probability of $\mathcal{O}$ can be represented as:

$$sky(\mathcal{O}) = 1 - \Pr(e_1) - \Pr(\bigcup_{i=2}^{n} e_i) + \Pr(e_1 \cap \bigcup_{i=2}^{n} e_i)$$

If $\mathcal{Q}_1 \prec \mathcal{O}$, by definition, it is easy to infer that $\mathcal{Q}_{i'} \prec \mathcal{O}$. In other words $\Pr(e_{i'}|e_1) = 1$. Therefore:

$$\Pr(e_1 \cap \bigcup_{i=2}^{n} e_i) = \Pr(e_1) \cdot \Pr(\bigcup_{i=2}^{n} e_i|e_1) = \Pr(e_1)$$

And the skyline probability of $\mathcal{O}$ is:

$$sky(\mathcal{O}) = 1 - \Pr(\bigcup_{i=2}^{n} e_i) = \Pr(\bigcap_{i=2}^{n} \overline{e_i})$$

□

Moreover, absorption satisfies transitivity, as stated in the following corollary:

COROLLARY 1. *Let $\mathcal{Q}_x \lhd \mathcal{Q}_y$ denote $\mathcal{Q}_y$ be absorbed by $\mathcal{Q}_x$. If $\mathcal{Q}_x \lhd \mathcal{Q}_y$ and $\mathcal{Q}_y \lhd \mathcal{Q}_z$. Then $\mathcal{Q}_x \lhd \mathcal{Q}_z$.*

PROOF. From $\mathcal{Q}_x \lhd \mathcal{Q}_y$ we infer that there must exist a set of dimensions, Γ_{xy}, where:

$$\begin{cases} \mathcal{Q}_x.j \neq \mathcal{O}.j, & j \in \Gamma_{xy} \\ \mathcal{Q}_x.j = \mathcal{Q}_y.j, & j \in \Gamma_{xy} \end{cases}$$

Similarly, from $\mathcal{Q}_y \lhd \mathcal{Q}_z$, we have:

$$\begin{cases} \mathcal{Q}_y.j \neq \mathcal{O}.j, & j \in \Gamma_{yz} \\ \mathcal{Q}_y.j = \mathcal{Q}_z.j, & j \in \Gamma_{yz} \end{cases}$$

Therefore, Γ_{xy} must be a subset of Γ_{yz} since at least on dimensions of Γ_{xy}, $\mathcal{Q}_y.j \neq \mathcal{O}.j$. This $\Gamma_{xy} \subset \Gamma_{yz}$ implies that $\mathcal{Q}_x.j = \mathcal{Q}_z.j$, $(j \in \Gamma_{xy})$. Consider $\mathcal{Q}_x.j \neq \mathcal{O}.j$, $(j \in \Gamma_{xy})$, we conclude that $\mathcal{Q}_x \lhd \mathcal{Q}_z$. □

Based on this transitivity property, during the absorption preprocessing, we only need to perform a one pass data

scan with arbitrary sequence to remove all unnecessary objects. That is we start from any $\mathcal{Q}_i$, then query a set of objects sharing same values with $\mathcal{Q}_i$ on all dimensions j, where $\mathcal{Q}_i.j \neq \mathcal{O}.j$. These objects can thus be safely 'absorbed' in computing $sky(\mathcal{O})$. Algorithm 3 presents details of this preprocessing procedure.

Algorithm 3: The absorption technique

Input: d-dimensional points $\mathcal{O}, \mathcal{Q}_1, ..., \mathcal{Q}_n$ in space D
Output: A set of objects that are really necessary in computing $sky(\mathcal{O})$

$\mathbb{S} \leftarrow \{\mathcal{Q}_1, \ldots, \mathcal{Q}_n\}$;
for *i=1 to n* **do**
  if $\mathcal{Q}_i$ *exists in* $\mathbb{S}$ **then**
    Compare $\mathcal{Q}_i$ and $\mathcal{O}$ to find a set of dimensions Γ, such that $\mathcal{Q}_i.j \neq \mathcal{O}.j$, $(j \in \Gamma)$;
    if $|\Gamma| \geq 1$ **then**
      Retrieve all points $\mathcal{Q}_{i'}$ such that $\mathcal{Q}_{i'}.j = \mathcal{Q}_i.j$, $(j \in \Gamma)$;
      Remove those $\mathcal{Q}_{i'}$ from $\mathbb{S}$
Return $\mathbb{S}$;

Partition. Recall that in the introduction section, we mentioned that dominance relations between two objects and $\mathcal{O}$ can be independent as long as those two objects share no common attribute values. Here we formally prove this independence property in Theorem 4.

THEOREM 4. *If n objects, $\mathcal{Q}_1, \ldots, \mathcal{Q}_n$, can be partitioned into m sets $\mathbb{S}_t$, $1 \leq t \leq m$ such that either no two objects from different sets share same attribute values, or only sharing same values as $\mathcal{O}$, $sky(\mathcal{O})$ can be simply computed as:*

$$sky(\mathcal{O}) = \prod_{t=1}^{m} \Pr(\bigcap_{\mathcal{Q}_i \in \mathbb{S}_t} \overline{e_i})$$

PROOF. Given m partitions, we have:

$$sky(\mathcal{O}) = \Pr(\bigcap_{t=1}^{m} (\bigcap_{\mathcal{Q}_i \in \mathbb{S}_t} \overline{e_i})) = 1 - \Pr(\bigcup_{t=1}^{m} C_t),$$

where $C_t = (\bigcup_{\mathcal{Q}_i \in \mathbb{S}_t} e_i)$.

Since no two objects from different sets share attribute values, it is easy to infer that given two sets $\mathbb{S}_t$ and $\mathbb{S}_{t'}$, we have:

$$\Pr(C_t \cap C_{t'}) = \Pr(C_t) \cdot \Pr(C_{t'})$$

Therefore we derive the skyline probability of $\mathcal{O}$ as:

$$\begin{aligned} sky(\mathcal{O}) = \; & 1 - \sum_{t=1}^{m} \Pr(C_t) + \sum_{t<t'} \Pr(C_t) \cdot \Pr(C_{t'}) \\ & + \cdots + (-1)^m \prod_{t=1}^{m} \Pr(C_t) \\ = \; & \prod_{t=1}^{m} (1 - \Pr(C_t)) \\ = \; & \prod_{t=1}^{m} \Pr(\bigcap_{\mathcal{Q}_i \in \mathbb{S}_t} \overline{e_i}) \end{aligned}$$

□

To avoid redundant preprocessing steps, we always apply absorption before partition. This guarantees that after partition, no more absorption procedures are necessary in every partitioned set. For instance, to compute $sky(\mathcal{O})$ in our running example (Figure 4), we first discard $\mathcal{Q}_1$ through absorption preprocessing. Then we partition remaining objects into three independent sets:

$$sky(\mathcal{O}) = \prod_{i=2}^{4} \Pr(\overline{e_i}) = \frac{3}{16}$$

Analyses. If objects follow a dense distribution, then absorption may remove a load of unnecessary objects. On the other hand, if objects take a really sparse distribution, more independent sets can be generated by partition. However there exist no performance guarantees for these two preprocessing techniques. Yet under certain data distribution these techniques may help exact solutions achieving polynomial time complexity, such as computing skyline probabilities on objects following block-zipf distribution, detailed in the next section.

6. EMPIRICAL STUDY

We evaluate our exact and approximate algorithms on real and synthetic data sets. In these data sets, we always assume uncertain preferences predefined between attribute values. The preference probabilities are randomly generated between $[0, 1]$, with 0 and 1 degenerating uncertain preferences to traditional certain ones. All algorithms are implemented in GNU C++ 4.3.2, and executed on a Linux 2.6.31.8 PC with 2.66 GHz Intel Xeon X3330 CPU and 4GB memory.

Synthetic data sets. We generate synthetic data sets in this empirical study for two testing purposes: a) evaluating the efficiency of our two preprocessing techniques. b) testing the accuracy and efficiency of our approximate algorithm on large data sets. In light of these, we create two types of data:

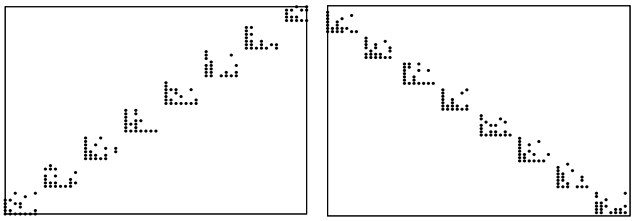
- Uniform: objects' attribute values are generated independently following uniform distributions on each dimension.
- Block-zipf: objects are grouped into several disjointed blocks where no two objects from different blocks share a common value. Inside each block, objects follow zipf's distribution with zipf parameter 1.

Note that unlike other skyline research papers in literature, we do not generate correlated/anti-correlated data sets. Because with uncertain preferences defined, a same block-zip data set can be correlated or anti-correlated with probabilities. Figure 8 shows correlated and anti-correlated block-zipf distributions.

Table 1 summarises parameters used in generating synthetic data sets.

Parameter	Range
Uniform data set cardinality (n)	10, 20, 40, 50
Block-zipf data set cardinality (n)	10, 1K, 10K, 100K
Dimensionality (d)	2, 3, 4, 5

Table 1: Parameter and ranges



(a) Correlated block-zipf (b) Anti-Correlated block-zipf

Figure 8: Block-zipf data distribution with different attribute value preferences

Real data sets. We also use a real data set publicly available from the UC Irvine Machine Learning Repository, called *Nursery* [1]. This data set contains 12,960 instances and 8 categorical attributes such as number of children, parents' occupation, etc.. Each instance in this data set represents an application submitted to the nursery school. And the nursery school will make decisions on these applications by ranking them on preferences over those categorical attributes. Obviously preferences on number of children can vary dramatically among various user perspectives. It is thus natural to model them with uncertain preferences. Because of missing detail preference information from the nursery school, without loss of generality, we generate synthetic preferences for those 8 attributes. Semantically, in this application, an instance's skyline probability is its possibility to be accepted by the school as a good application.

Algorithms. Table 2 lists all four implemented algorithms, with two exact ones and two approximate ones. Det is the deterministic algorithm, i.e. Algorithm 1. Det+ preprocesses data with two speed-up techniques, i.e. absorption and partition, before applying Det on skyline probability computations. Similarly, Sam is the Monte Carlo sampling algorithm, i.e. Algorithm 2. Sam+ applies our preprocessing techniques before running Sam on data sets.

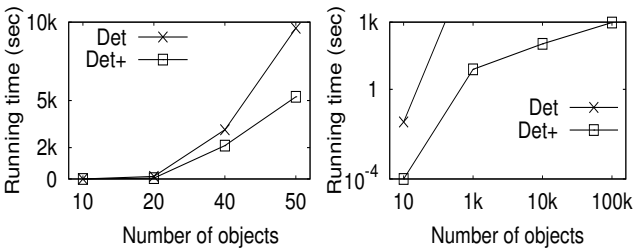
Algorithm	Abbreviation
Deterministic	Det
Deterministic with data preprocessing	Det+
Monte Carlo sampling	Sam
Sampling with data preprocessing	Sam+

Table 2: Algorithms and their abbreviations

Performance metrics. We use average running time and absolute errors to measure performance of our four algorithms. Specifically, if a data set has no more than 1000 objects, we will calculate every object's skyline probability and then compute average values. Otherwise, we will randomly pick 1000 objects to compute average values.

6.1 Deterministic Algorithms

We evaluate performances of our two deterministic algorithms on the uniform and block-zipf data sets. First, we fix the data dimensionality as 5 and vary the size of data sets from 10 to 50 on uniform data, and 10 to 100K on block-zipf data. Figure 9 reports efficiency comparisons between Det and Det+ on different data sets. In Figure 9 (a), it is not



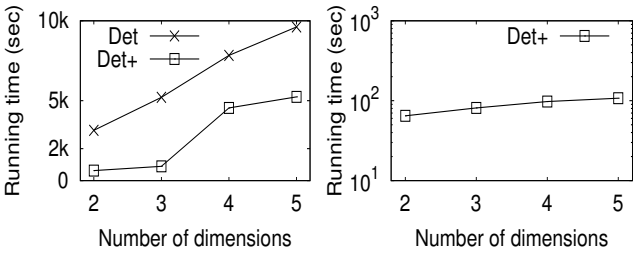
(a) Uniform (5d) (b) Block-zipf (5d)

Figure 9: Efficiency of exact algorithms (varying n)

surprising to see both deterministic algorithms can not compute skyline probabilities of a uniform data set with more than 50 objects, within 10^4 seconds. However, although preprocessing add some overheads to Det+, we notice that the overall performance of Det+ is still much better than Det, due to removal of unnecessary objects in its computation.

Moreover, in Figure 9 (b),while Det's performance on the block-zipf data is similar to that on a uniform data set, Det+ can efficiently compute skyline probability of a data set with up to 100k objects, within 1000 seconds. This demonstrates the efficiency of our preprocessing techniques.

Similar conclusions can be drawn from Figure 10, where we fix the size of data sets as 50/10k accordingly and vary dimensionality of data sets. It is interesting to note that Det+ performs especially well on low dimensional data, due to 'absorption' efficiently removing many unnecessary objects in preprocessing. Also note that in Figure 10 (b), we only report performance of Det+ since Det can not compute any object's skyline probability in these data sets within 10^4 seconds.



(a) Uniform (50) (b) Block-zipf (10k)

Figure 10: Efficiency of exact algorithms (varying d)

6.2 Approximate algorithms

We evaluate the accuracy and performance of our two approximate algorithms on the uniform and block-zipf data sets. For both algorithms, we set $\epsilon = 0.01$, $\delta = 0.01$ and the sample size as 3000. Note that although theoretically the sample size for both algorithms should be 26492 ($\frac{1}{2\epsilon^2} \ln \frac{2}{\delta}$). From our empirical studies on running Sam and Sam+ on a block-zipf 5-d data set with 100k objects, we observe that 3000 is already a good enough sample size that satisfies the $\epsilon = 0.01$ error bound. Figure 11 reports the results of our

[1] http://archive.ics.uci.edu/ml/datasets/Nursery

empirical studies with different absolute approximation errors generated by Sam and Sam+ under various sample sizes.

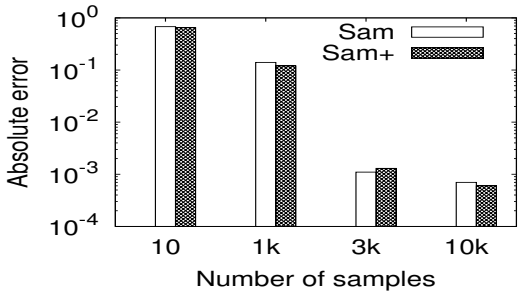
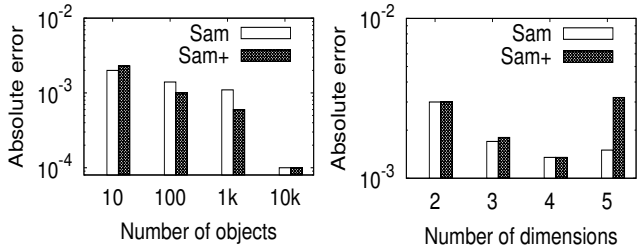


Figure 11: Absolute errors with various sample sizes

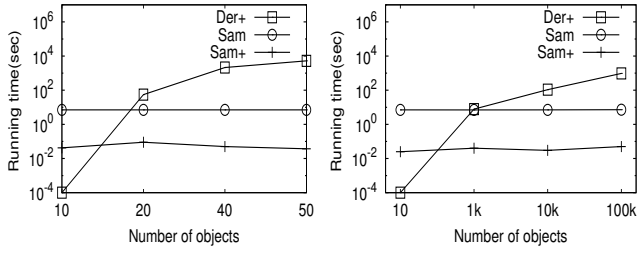
We compare approximate results by Sam and Sam+ against exact ones of block-zipf data sets in Figure12. Both algorithms output approximations with absolute errors well below $\epsilon = 0.01$.



(a) 5-d objects (b) 10k objects

Figure 12: Approximation accuracy with $\epsilon = \delta = 0.01$

Similar as efficiency evaluations on deterministic algorithms, we compare efficiency of Sam and Sam+ on various synthetic data sets. We also include reported running time of Det+ as a reference. Figure 13 compares efficiency of Sam and Sam+, on uniform and block-zipf data sets with 5-d objects. Note that due to the logarithmic scale used in this figure, the same Det+ looks different to that displayed in Figure 9 (a).

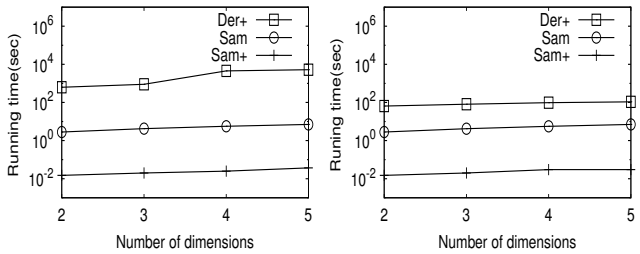


(a) Uniform (5d) (b) Block-zipf (5d)

Figure 13: Efficiency of approx. algorithms (varying n)

It is interesting to note that on a small data set or under a certain data distribution (block-zipf in this paper), Det+ performs even better than sampling algorithms, if not always. However when data sets get larger, sampling algorithms undoubtedly beat their deterministic opponents.

Figure 14 reports similar performance comparison results on fixed-size data sets with varying dimensionality.

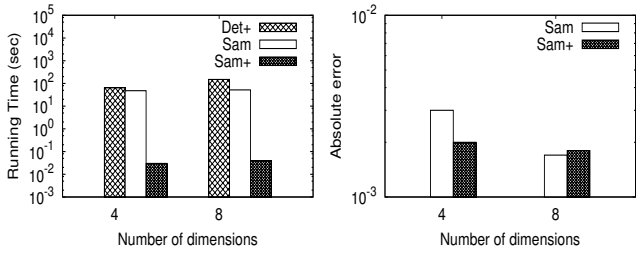


(a) Uniform (50) (b) Block-zipf (10k)

Figure 14: Efficiency of approx. algorithms (varying d)

6.3 Evaluation on real data sets

Finally we evaluate our deterministic and approximate algorithms on the nursery school data set, which contains 12960 8-dimensional objects. From it, we generate two data sets with dimensionality equals to 4 and 8 correspondingly. Figure 15 reports the experiment results. Note that since Det can not deliver any object's skyline probability results within 10^4 seconds, we omit it in Figure 15. Also note that although the worst case time complexity of Det+ is exponential, it still can efficiently compute skyline probabilities of objects in these real data sets because of the preprocessing techniques.



(a) Efficiency comparison (b) Error comparison

Figure 15: Evaluation on real data sets

7. RELATED WORK

Previously known as Pareto sets [2] or maximum vectors [10], the skyline operator was first introduced in the database community by Börzsönyi et al. to support multi-criteria decision making [4]. Aiming at efficient skyline query processing on very large data sets, many techniques have been proposed thereafter, ranging from intuitive block nested loop approach, to intricate index-based skyline query processing methods [5, 22, 18, 6, 3, 11]. Variations of skyline query have also emerged from literature, such as skycube [24], reverse skyline [12], sliding window skyline [14, 23], approximate skyline [9] etc. Note that the approximate skyline in [9] refers to finding a subset of representative skyline points, which should not be confused with the probabilistic skyline later appears in literature.

Pei et al. first investigated skyline computation problems on uncertain data [19]. In their study, attribute values of an

object follow random distributions. Therefore dominance relations between objects become uncertain. Consequently every object will have a probability to be a skyline point. To compute the probabilistic skyline is thus to find all objects having their skyline probabilities greater than a threshold τ. Motivated by this uncertain data model, many variations of probabilistic skyline have been investigated, such as probabilistic reverse skyline [1, 13], probabilistic top-k skyline query [25], etc.

Most of those probabilistic skyline models assume uncertain attribute values. However, uncertain preferences are also very common in real life. Various uncertain preference models have been well discussed in [26], such as those used in voting systems. Sacharidis et. al first studied an uncertain preferences based probabilistic skyline model [21], which was also referred as a fuzzy skyline model in [7]. Nevertheless compared to the crowded research activities of probabilistic skyline over uncertain values, finding probabilistic skyline over uncertain preferences seems very quiet. To the extend of our knowledge, [21] is the only paper that formally defines probabilistic skyline over uncertain preferences, and provides a first solution as discussed before.

8. CONCLUSIONS

In this paper, we revisited the skyline probability computation problem over uncertain preferences. We proved the ♯P-completeness of this problem and presented deterministic as well as randomised approaches, with two speed-up preprocessing techniques. Using synthetic and real data sets, the empirical studies demonstrated both efficiency and effectiveness of our algorithms.

Unlike computing probabilistic skylines over uncertain data where many efficient polynomial algorithms have already been developed, computing probabilistic skyline over uncertain preferences becomes extremely hard given the computational intractability of even computing one single object's skyline probability. A naïve approach will be calculating every object's skyline probability by applying the sampling algorithm proposed in this paper. However a more efficient solution may be applying the generic top-k evaluation framework for uncertain databases, proposed in [20]. This will be one of our future works.

9. REFERENCES

[1] M. Bai, J. Xin, and G. Wang. Probabilistic reverse skyline query processing over uncertain data stream. In *Proceedings of the 17th international conference on Database Systems for Advanced Applications - Volume Part II*, DASFAA'12, pages 17–32, Berlin, Heidelberg, 2012. Springer-Verlag.

[2] O. Barndorff-Nielsen and M. Sobel. On the distribution of the number of admissible points in a vector random sample. *Theory of Probability and its Application*, 11(2):249–269, 1966.

[3] I. Bartolini, P. Ciaccia, and M. Patella. Efficient sort-based skyline evaluation. *ACM Trans. Database Syst.*, 33(4):1–49, 2008.

[4] S. Borzsony, D. Kossmann, and K. Stocker. The Skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 421–430, 2001.

[5] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with presorting. In *ICDE*, pages 717–816, 2003.

[6] P. Godfrey, R. Shipley, and J. Gryz. Algorithms and analyses for maximal vector computation. *The VLDB Journal*, 16(1):5–28, 2007.

[7] A. Hadjali, O. Pivert, and H. Prade. On different types of fuzzy skylines. In *Proceedings of the 19th international conference on Foundations of intelligent systems*, ISMIS'11, pages 581–591, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] A. Kangas, J. Kangas, and M. Kurttila. *Decision Support for Forest Management (Managing Forest Ecosystems)*. Springer, softcover reprint of hardcover 1st ed. 2008 edition, Nov. 2010.

[9] V. Koltun and C. H. Papadimitriou. Approximately dominating representatives. *Theor. Comput. Sci.*, 371(3):148–154, 2007.

[10] H. Kung and F. Luccio. On finding the maxima of a set of vectors. *Journal of the ACM (JACM)*, 1975.

[11] K. C. K. Lee, B. Zheng, H. Li, and W.-C. Lee. Approaching the skyline in z order. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 279–290. VLDB Endowment, 2007.

[12] X. Lian and L. Chen. Monochromatic and bichromatic reverse skyline search over uncertain databases. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 213–226, New York, NY, USA, 2008. ACM.

[13] X. Lian and L. Chen. Reverse skyline search in uncertain databases. *ACM Trans. Database Syst.*, 35(1):3:1–3:49, Feb. 2008.

[14] X. Lin, Y. Yuan, W. Wang, and H. Lu. Stabbing the sky: Efficient skyline computation over sliding windows. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering*, pages 502–513, Washington, DC, USA, 2005. IEEE Computer Society.

[15] M. Mitzenmacher and E. Upfal. *Probability and Computing*. The press syndicate of he university of cambridge, 2005.

[16] R. Motwani. *Randomized Algorithms*. 1995.

[17] D. Mueller. Probabilistic majority rule. *Kyklos*, 1989.

[18] D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. *ACM Trans. Database Syst.*, 30(1):41–82, 2005.

[19] J. Pei, B. Jiang, X. Lin, and Y. Yuan. Probabilistic skylines on uncertain data. In B. Jiang, editor, *Proceedings of the 33rd international conference on Very large data bases*, pages 15–26. VLDB Endowment, 2007.

[20] C. Re, N. Dalvi, and D. Suciu. Efficient Top-k Query Evaluation on Probabilistic Data. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 886–895, 2007.

[21] D. Sacharidis, A. Arvanitis, and T. Sellis. Probabilistic contextual skylines. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pages 273–284, 2010.

[22] K.-L. Tan, P.-K. Eng, and B. C. Ooi. Efficient

progressive skyline computation. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 301–310, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[23] Y. Tao and D. Papadias. Maintaining sliding window skylines on data streams. *IEEE Trans. on Knowl. and Data Eng.*, 18(3):377–391, 2006.

[24] Y. Yuan, X. Lin, Q. Liu, W. Wang, J. X. Yu, and Q. Zhang. Efficient computation of the skyline cube. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 241–252. VLDB Endowment, 2005.

[25] Y. Zhang, W. Zhang, X. Lin, B. Jiang, and J. Pei. Ranking uncertain sky: The probabilistic top-k skyline operator. *Information Systems*, 36(5):898–915, 2011.

[26] M. ÃŰztÃijrk and A. TsoukiÃăs. Preference modelling. In *State of the Art in Multiple Criteria Decision Analysis*, pages 27–72. Springer-Verlag, 2005.