Getting the Best from Uncertain Data

Ilaria Bartolini, Paolo Ciaccia, and Marco Patella
DEIS - Università di Bologna, Italy
{i.bartolini,paolo.ciaccia,marco.patella}@unibo.it

Abstract. The skyline of a relation is the set of tuples that are not dominated by any other tuple in the same relation, where tuple u dominates tuple v if u is no worse than v on all the attributes of interest and strictly better on at least one attribute. Previous attempts to extend skyline queries to probabilistic databases have proposed either a weaker form of domination, which is unsuitable to univocally define the skyline, or a definition that implies algorithms with exponential complexity. In this paper we demonstrate how, given a semantics for linearly ranking probabilistic tuples, *the skyline of a probabilistic relation can be univocally defined*. Our approach preserves the three fundamental properties of skyline: 1) it equals the union of all top-1 results of monotone scoring functions, 2) it requires no additional parameter to be specified, and 3) it is insensitive to actual attribute scales. We also detail efficient sequential and index-based algorithms.

1 Introduction

Uncertain data management has recently become a very active area of research, due to the huge number of relevant applications in which uncertainty plays a key role, such as data extraction from the Web, data integration, biometric systems, sensor network readings, etc. Further, uncertainty might also occur as a result of data anonymization.

According to a commonly adopted model, uncertain data can be represented through *probabilistic relations*, in which each tuple has also a probability (confidence) to appear [11, 12]. A probabilistic relation compactly represents a set of *possible worlds*, i.e., subsets of tuples. In the general case, the formation of possible worlds is constrained by a set of generation rules, that are used to model correlation among tuples (e.g., a rule might state that two tuples are mutually exclusive).

In recent years, several works have focused on extending different query types to probabilistic databases. Among them, in this paper we concentrate on *skyline queries*, whose relevance in supporting multi-criteria decision analysis is well known [3]. The skyline of a relation R is the set of undominated (or Pareto-optimal) tuples in R, where tuple u dominates tuple v if u is no worse than v on all the attributes of interest, and strictly better than v on at least one attribute. The appeal of skyline queries comes from the observation that the skyline consists of all and only top-1 results obtainable from scoring functions that are monotone in the skyline attributes, thus providing users with an overall picture of what are the best alternatives in a relation. Further, unlike top-k queries, a skyline query does not require any input parameter to be specified. Not less important

is also the fact that the skyline is insensitive to attributes' scales, being it only dependent on the relative ordering of tuples on each attribute.

As a motivating example, consider a traffic-monitoring application collecting data by means of a radar, a sample of which is shown in Figure 1.[1] Each radar reading has associated a Prob value, representing the overall confidence one has in the reading itself. A skyline query on the Time and Speed attributes would

TID	Plate No	Time	Speed	Prob
t_1	X-123	11:50	145	0.4
t_2	W-246	11:40	160	0.3
t_3	Z-456	11:15	145	0.6
t_4	H-121	11:05	137	0.4
t_5	Y-324	11:00	140	0.6
t_6	X-827	10:50	135	0.4
t_7	C-442	10:45	155	0.5

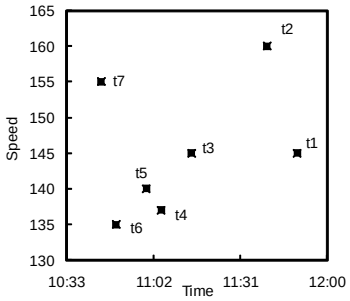


Fig. 1. A probabilistic relation

return those tuples (i.e., readings) that are, at the same time, the most recent ones and that concern high-speed cars. In the deterministic case it would be $\text{SKY}(R) = \{t_1, t_2\}$, as it can be easily verified from the figure on the right. In the probabilistic case, in which also the confidence of each tuple has to be considered, even *defining* what the skyline should be is challenging.

1.1 Related Work

The first work to consider skyline queries on probabilistic data has been [10]. There, the basic idea is to compute for each tuple u the probability, $\Pr_{\text{SKY}}(u)$, that u is undominated, and then rank tuples based on these *skyline probabilities*. Intuitively, $\Pr_{\text{SKY}}(u)$ equals the overall probability of the possible worlds W in which u is in the (deterministic) skyline of W. The p-skyline of a probabilistic relation is then defined as the set of tuples whose skyline probability is at least p. This approach is unable to preserve the basic skyline properties, since it requires an additional parameter (the p threshold), and has no apparent relationship with the results of top-1 queries. Subsequent works on the subject have provided efficient algorithms to compute all skyline probabilities [1], and shown how to compute p-skylines on uncertain data streams [14]. More recently, Lin et al. have proposed the *stochastic skyline operator* [9]. Unlike p-skyline, the stochastic skyline has the advantage of not requiring any parameter. However, this comes at the price of an algorithmic exponential complexity, since testing stochastic domination is an NP-complete problem. Further, the stochastic skyline equals only a subset of possible top-1 results, namely those arising from the *expectation of multiplicative scoring functions*.

1.2 Contributions

In this paper we address the problems of *defining and efficiently computing the skyline of a probabilistic relation*. We start by providing in Section 3 a formal

[1] A similar example was also used in previous works on top-k queries [12, 8].

definition of skyline, which is based on a generalization to the probabilistic case of the concept of domination among tuples. The P-domination relationship we introduce to this purpose is formally grounded in order theory, and satisfies all the properties the skyline has in the deterministic case. Since P-domination is parametric in the semantics used to rank probabilistic tuples, this implies that, whatever ranking semantics for top-k queries one wants to adopt, our skyline definition will be always consistent with it, which is a remarkable property.

In Section 4 we show how the skyline can be computed in $\mathcal{O}(n^3)$ time for a relation with n tuples, by detailing the analysis for the case in which the "expected rank" semantics is used for ranking tuples [5]. In Section 5 we describe algorithms aiming to reduce the actual response time. Experimental evaluation on large datasets shows the practical applicability of our approach.

For lack of space, we only consider probabilistic relations in which tuples are pairwise independent, i.e., no generation rule is present; however, our results can be also smoothly extended to the correlated case.

2 Preliminaries

We model a probabilistic relation R^p as a pair, $R^p = (R, p)$, where R is a relation in the standard sense, also called a *deterministic* relation, and p is a function that assigns to each tuple $u \in R$ a probability, $p(u) \in (0, 1]$. A *posssible world* W of R^p is any subset of tuples from R. The set of possible worlds of R^p is denoted $\mathcal{W}$. The probability of possible world W is computed as: $\Pr(W) = \prod_{u \in W} p(u) \prod_{v \notin W} (1 - p(v))$.

Given a (deterministic) relation R whose schema includes a set of numerical attributes $\mathcal{A} = \{A_1, A_2, \ldots, A_d\}$, the *skyline* of R with respect to $\mathcal{A}$, denoted $\textsc{Sky}_{\mathcal{A}}(R)$ or simply $\textsc{Sky}(R)$, is the set of *undominated* tuples in R. Assuming that on each attribute higher values are preferable, tuple u (Pareto-)dominates tuple v, written $u \succ v$, iff it is $u.A_i \geq v.A_i$ for each $A_i \in \mathcal{A}$ and there exists at least one attribute A_j such that $u.A_j > v.A_j$. Thus:

$$\textsc{Sky}(R) = \{u \in R \mid \nexists\, v \in R : v \succ u\} \tag{1}$$

If neither $u \succ v$ nor $v \succ u$ hold, then u and v are *indifferent*, written $u \sim v$.

A *scoring function* $s()$ on the attributes $\mathcal{A}$, $s : dom(\mathcal{A}) \to \Re$, is *monotone* iff $u.A_i \geq v.A_i$ $(i = 1, \ldots, d)$ implies $s(u) \geq s(v)$. Although it is folklore that $\textsc{Sky}(R)$ equals the union of top-1 results of monotone scoring functions, this is imprecise because of the non-deterministic nature of top-1 queries. For instance, consider the max function, which is monotone, and $R = \{(3, 4), (1, 4)\}$. Although $(3, 4) \succ (1, 4)$, it is $\max\{3, 4\} = \max\{1, 4\}$, thus $(1, 4)$ might be (non-deterministically) returned as the top-1 result. To obviate the problem, in this paper we only consider monotone functions that are also *domination-preserving*, i.e., $u \succ v$ implies $s(u) > s(v)$.[2] In the following, we always implicitly assume that a monotone function is also domination-preserving.

[2] Domination-preserving monotone functions are exactly those functions that Fagin et al. call *strictly monotone in each argument* [6].

3 The Skyline of a Probabilistic Relation

In order to define the skyline of a probabilistic relation we start by rewriting Equation 1 as:

$$\boxed{\textsc{Sky}(R^p) = \{u \in R \mid \nexists\, v \in R : v \succ_p u\ \}} \tag{2}$$

in which the only difference with the deterministic case is that $\succ$ is substituted by $\succ_p$. We call $\succ_p$ probabilistic domination, or *P-domination* for short. Note that $\succ_p$ is a binary relation in the standard sense, i.e., no probability is present in $\succ_p$.

In order to define P-domination so as to preserve all skyline properties, we approach the problem by considering things from an order-theoretic viewpoint. In order-theoretic terms, $\succ$ is a *strict partial order*, i.e., an irreflexive ($\forall u : u \not\succ u$) and transitive ($\forall u, v, t : u \succ v \wedge v \succ t \Rightarrow u \succ t$) relationship on the domain of skyline attributes. A *linear* order $\succcurlyeq$ is a strict partial order that is also *connected*, i.e., for any two distinct tuples u and v, either $u \succcurlyeq v$ or $v \succcurlyeq u$.[3] A linear order $\succcurlyeq$ is called a *linear extension* of $\succ$ iff $u \succ v \Rightarrow u \succcurlyeq v$, i.e., $\succcurlyeq$ is *compatible* with $\succ$. Notice that a linear extension of $\succ$ can be obtained by ordering tuples with a monotone scoring function and then breaking ties arbitrarily.

Let $\textsc{Ext}(\succ)$ denote the set of all linear extensions of $\succ$. A fundamental result in order theory, derived from *Szpilrajn's Theorem* [13], asserts that *any strict partial order $\succ$ equals the intersection of its linear extensions*, $\succ = \bigcap\{\succcurlyeq \mid \succcurlyeq \in \textsc{Ext}(\succ)\}$. This is the first ingredient needed to define P-domination.

Our second ingredient comes from the observation that *each linear order $\succcurlyeq$ on the tuples of R can be used to define a corresponding linear order on the probabilistic tuples of R^p*. Indeed, this has been the subject of several recent works aiming to support top-k queries on uncertain data, which has lead to different, alternative semantics for ranking tuples that come with both a score and a probability [12, 15, 5]. In abstract terms, each of these semantics can be viewed as a *probabilistic ranking function* Ψ that, given a linear order $\succcurlyeq$ on the tuples of R and a probability function p, yields a *probabilistic linear order* $\succcurlyeq_p = \Psi(\succcurlyeq, p)$ on the probabilistic tuples of R^p. In practice, any ranking semantics assigns to each tuple u a value $\psi(u)$, so that $u \succcurlyeq_p v$ iff $\psi(u) > \psi(v)$.[4]

We are now ready to define P-domination:

Definition 1 (P-domination). *Let $R^p = (R, p)$ be a probabilistic relation, and let $\succ$ be the Pareto-domination relationship on the tuples in R when considering the skyline attributes $\mathcal{A}$. Let Ψ be a probabilistic ranking function on R^p. For any two tuples u and v in R^p, we say that u* P-dominates *v, written $u \succ_p v$, iff for each linear extension $\succcurlyeq$ of $\succ$, with associated probabilistic linear order $\succcurlyeq_p = \Psi(\succcurlyeq, p)$, it is $u \succcurlyeq_p v$, that is:*

$$\boxed{u \succ_p v \iff u \succcurlyeq_p v,\ \ \forall\ \succcurlyeq_p = \Psi(\succcurlyeq, p), \succcurlyeq \in \textsc{Ext}(\succ)} \tag{3}$$

[3] To denote linear orders over tuples we use the symbol $\succcurlyeq$ in place of the usual $>$, and reserve the latter for the standard order on real numbers.

[4] In the most general case, Ψ might also depend on the actual scores of the tuples, rather that only on their ordering. This has no influence on the results we derive.

The diagram in Figure 2 summarizes how $\succ_p$ is conceptually obtained: from $\succ$ we obtain a set of linear orders, and for each of them a corresponding probabilistic linear order. The intersection of such probabilistic rankings yields P-domination.

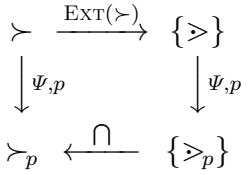


Fig. 2. How P-domination is obtained

From Definition 1 three major results follow:[5]

Theorem 1. *For any probabilistic ranking function Ψ, the corresponding P-domination relationship $\succ_p$ is a strict partial order.*

Theorem 2. *Let $\text{Sky}(R^p)$ be the skyline of R^p, for a given probabilistic ranking function Ψ. A tuple u belongs to $\text{Sky}(R^p)$ iff there exists a monotone scoring function $s()$ such that u is the top-1 tuple according to the probabilistic linear order $\succcurlyeq_p = \Psi(\succcurlyeq, p)$, where $\succcurlyeq$ is the linear order induced by $s()$ on R.*

A further important property of $\text{Sky}(R^p)$ is that, as in the deterministic case, it is insensitive to actual attribute values, rather it only depends on the relative ordering on each skyline attribute.

Theorem 3. *Let $R^p = (R, p)$ be a probabilistic relation, and $S^p = (S, p)$ be another probabilistic relation, in which S is obtained from R through an isomorphism ϕ that preserves Pareto domination (i.e., for any two tuples $u, v \in R$ it is $u \succ v$ if and only if $\phi(u) \succ \phi(v)$), and $p(u) = p(\phi(u))$ for all $u \in R$. Then, for any probabilistic ranking function Ψ, it is $\text{Sky}(R^p) = \text{Sky}(S^p)$.*

4 Computing P-domination

Definition 1 cannot be directly used to check P-domination, since it requires to enumerate all linear extensions of the Pareto dominance relationship, and these can be exponential in the number of tuples.[6] In the following we first sketch how, independently of the specific probabilistic ranking function Ψ, P-domination can be checked without materializing the linear extensions of $\succ$, after that we detail the analysis for the case of in which Ψ is the "expected rank" semantics [5].

Consider a linear extension $\succcurlyeq$ of $\succ$, and let $\psi_{\succcurlyeq}(u)$ be the numerical value that Ψ assigns to tuple u. According to Definition 1, for $u \succ_p v$ to hold it has to be $\psi_{\succcurlyeq}(u) > \psi_{\succcurlyeq}(v)$ for all linear extensions $\succcurlyeq$ of $\succ$, that is:

$$u \succ_p v \iff \min_{\succcurlyeq \in \text{Ext}(\succ)} \left\{ \frac{\psi_{\succcurlyeq}(u)}{\psi_{\succcurlyeq}(v)} \right\} > 1 \tag{4}$$

[5] For lack of space, all formal results are stated without proof.

[6] If R^p consists of n pairwise indifferent tuples, then $\succ$ is empty and $\text{Ext}(\succ)$ has size $n!$, since each permutation is compatible with $\succ$.

The key idea for efficiently checking the above inequality is to determine which is the linear order that is the most unfavorable one for u with respect to v. If $\psi_{\succcurlyeq}(u) > \psi_{\succcurlyeq}(v)$ holds for this "extremal" order, then it will necessarily hold for all other orders compatible with $\succ$. Regardless of the specific probabilistic ranking function Ψ, the two relevant cases to consider here are:

$u \succ v$: When u dominates v, we can restrict the analysis to those linear orders for which it is $u \succcurlyeq v$; starting from this we analyze how other tuples should be arranged in the linear order so as to minimize the ratio $\psi_{\succcurlyeq}(u)/\psi_{\succcurlyeq}(v)$.

$u \not\succ v$: If u does not dominate v, then the worst case for u and the best one for v corresponds to a linear order in which: 1) $u \succcurlyeq t$ only for those tuples t that u dominates, and 2) $t' \succcurlyeq v$ only for those tuples t' that dominate v.

4.1 P-domination with Expected Ranks

According to [5], the result of a top-k query on a probabilistic relation R^p is based on the concept of *expected rank*. Given a linear order $\succcurlyeq$ on the tuples of R, the rank of u in a possible world W with $|W|$ tuples is the number of tuples in W that precedes u, that is:

$$\text{rank}_{W,\succcurlyeq}(u) = \begin{cases} |\{t \in W \mid t \succcurlyeq u\}| & \text{if } u \in W \\ |W| & \text{otherwise} \end{cases}$$

Thus, ranks range from 0 to $|W| - 1$, and tuples not in W have rank $|W|$. The expected rank of u is then defined as $ER_{\succcurlyeq}(u) = \sum_{W \in \mathcal{W}} \text{rank}_{W,\succcurlyeq}(u) \times \Pr(W)$.

As in [5], we consider that if two tuples have a same expected rank value, a tie-breaking rule is applied so that expected ranks define a linear order. Let $\succcurlyeq_p$ be such linear order, i.e.,: $u \succcurlyeq_p v$ iff $ER_{\succcurlyeq}(u) < ER_{\succcurlyeq}(v)$.

As explained in [5], the expected rank of a tuple u can be computed as:

$$ER_{\succcurlyeq}(u) = p(u) \times \sum_{t \succcurlyeq u} p(t) + (1 - p(u)) \times \sum_{t \neq u} p(t) \tag{5}$$

where the first term is the expected rank of u in a possible world in which u appears, and the second sum is the expected size of a possible world in which u does not appear.

Let P be the overall probability of all the tuples in R, $P = \sum_{t \in R} p(t)$, and let $H_{\succcurlyeq}(u) = \sum_{t \succcurlyeq u} p(t)$ be the overall probability of those tuples that are better than u according to $\succcurlyeq$. A key observation that will be exploited in the following is that, for any linear order $\succcurlyeq$ that extends $\succ$, it is $H_{\succcurlyeq}(u) \in [H^-(u), H^+(u)]$, where the two bounds are respectively defined as:

$$H^-(u) = \sum_{t \succ u} p(t) \qquad H^+(u) = \sum_{\substack{u \not\succ t \\ t \neq u}} p(t) = P - p(u) - \sum_{u \succ t} p(t)$$

Notice that $H^-(u)$ is the best possible case for u, in which only those tuples that dominate u are also better than u according to $\succcurlyeq$, whereas the worst possible

case for u is given by a linear order in which u is better only of those tuples that it dominates. Equation 5 can then be compactly rewritten as:

$$ER_{\succcurlyeq}(u) = p(u) \times H_{\succcurlyeq}(u) + (1 - p(u)) \times (P - p(u))$$

According to Definition 1, it has to be $ER_{\succcurlyeq}(u) < ER_{\succcurlyeq}(v)$ for each linear order $\succcurlyeq$ that extends $\succ$, i.e.:

$$\max_{\succcurlyeq \in \text{EXT}(\succ)} \left\{ \frac{p(u) \times H_{\succcurlyeq}(u) + (1 - p(u)) \times (P - p(u))}{p(v) \times H_{\succcurlyeq}(v) + (1 - p(v)) \times (P - p(v))} \right\} < 1$$

Let $P_{u,v} = P - p(u) - p(v)$. Substituting, simplifying, and rearranging terms, above inequality can be equivalently written as:

$$u \succ_p v \Leftrightarrow \boxed{\frac{p(u)}{p(v)} > \max_{\succcurlyeq \in \text{EXT}(\succ)} \left\{ \frac{P_{u,v} + 1 - H_{\succcurlyeq}(v)}{P_{u,v} + 1 - H_{\succcurlyeq}(u)} \right\}} \tag{6}$$

The two cases to be considered for Equation 6 are dealt with as follows.

$u \succ v$: Since u dominates v, and domination is transitive, it is $H_{\succcurlyeq}(v) \geq H_{\succcurlyeq}(u) + p(u)$ for each $\succcurlyeq \in \text{EXT}(\succ)$. This ensures that the right-hand side of Equation 6 is strictly less than 1, which immediately yields the first P-domination rule:

$$\boxed{u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq 1} \tag{Rule 1}$$

Note that this perfectly matches the intuition that a more likely and better tuple should probabilistically dominate a less likely and worse tuple.

When $p(u) < p(v)$, we can maximize the right-hand side of Equation 6 as follows. For any tuple t such that $u \succ t$, yet t is indifferent to v, $t \sim v$, we set $v \succcurlyeq t$, so as not to increase the value of $H_{\succcurlyeq}(v)$. For a tuple t which is indifferent to both u and v there are two alternatives to consider: either $t \succcurlyeq u \succcurlyeq v$ or $u \succcurlyeq v \succcurlyeq t$. In the first case we would add $p(t)$ to both $H_{\succcurlyeq}(v)$ and $H_{\succcurlyeq}(u)$, but this would *lower* the ratio in the right-hand side of Equation 6. Thus, we conclude that the second alternative is the one to be chosen. Finally, consider a tuple t such that $t \succ v$, yet $t \sim u$. In this case we set $t \succcurlyeq u$, so as to increase the value of $H_{\succcurlyeq}(u)$ (notice that $H_{\succcurlyeq}(v)$ already includes $p(t)$, since $t \succ v$).

Combining the above cases, it is evident that it is $H_{\succcurlyeq}(v) = H^-(v)$. On the other hand, for $H_{\succcurlyeq}(u)$ we have to add to $H^-(u)$ the mass of probability of all those tuples t such that $t \succ v$ and $t \sim u$, that is: $H_{\succcurlyeq}(u) = H^-(u) + \sum_{\substack{t \succ v \\ t \sim u}} p(t)$. By partitioning the set of tuples that dominate v depending on their relationship with respect to u, the following identity is derived:

$$H^-(v) = H^-(u) + p(u) + \sum_{\substack{t \succ v \\ t \sim u}} p(t) + \sum_{\substack{t \succ v \\ u \succ t}} p(t)$$

Letting $IbP(u, v) = \sum_{u \succ t \succ v} p(t)$ to stand for the *in-between mass of probability* of those tuples that dominate v and are dominated by u we obtain:

$$H_{\succcurlyeq}(u) = H^-(v) - IbP(u, v) - p(u)$$

from which we get the second P-domination rule:

$$\boxed{u \succ v \ \wedge \ \frac{p(u)}{p(v)} > \frac{P_{u,v} + 1 - H^-(v)}{P_{u,v} + 1 - H^-(v) + IbP(u,v) + p(u)}} \qquad \text{(Rule 2)}$$

Rule 2 generalizes Rule 1, which is therefore redundant. However we keep it since, unlike Rule 2, it can be checked without the need to compute any bound.

u $\not\succ$ v: P-domination can occur even when $u \not\succ v$, provided $p(u) > p(v)$. In this case it is immediate to see that the right-hand side of Equation 6 is maximized by setting $H_{\succcurlyeq}(v) = H^-(v)$ and $H_{\succcurlyeq}(u) = H^+(u)$, thus:

$$\boxed{u \not\succ v \ \wedge \ \frac{p(u)}{p(v)} > \frac{P_{u,v} + 1 - H^-(v)}{P_{u,v} + 1 - H^+(u)}} \qquad \text{(Rule 3)}$$

Example 1. Table 1 lists the probabilities of the tuples in Figure 1, whose overall probability is $P = 3.2$, together with their H^- and H^+ bounds. As an example of how bounds are computed consider tuple t_3. Since t_3 is dominated only by t_1 and t_2, it is $H^-(t_3) = p(t_1) + p(t_2) = 0.7$. The tuples dominated by t_3 are t_4, t_5, and t_6, thus $H^+(t_3) = P - p(t_3) - p(t_4) - p(t_5) - p(t_6) = 1.2$. A case to which Rule 1 applies concerns tuples t_1 and t_4, since it is $t_1 \succ t_4$ and $p(t_1) \geq p(t_4)$, thus $t_1 \succ_p t_4$. Rule 2 is used to discard tuple t_5, which is P-dominated by t_1 (notice that here it is $p(t_1) < p(t_5)$, and $IbP(t_1, t_5) = 0.6$). A case in which Rule 3 is satisfied regards tuples t_3 and t_2 (notice that t_2 is part of the deterministic skyline). An exhaustive analysis shows that $\text{SKY}(R^p) = \{t_1, t_3, t_7\}$. □

tuple	t_1	t_2	t_3	t_4	t_5	t_6	t_7
probability	0.4	0.3	0.6	0.4	0.6	0.4	0.5
H^-	0	0	0.7	1.3	1.3	2.3	0.3
H^+	0.8	0.4	1.2	2.4	2.2	2.8	2.7

Table 1. Probabilities and bounds for the dataset in Figure 1

5 Algorithms

The skyline of a probabilistic relation R^p consisting of n tuples can be computed in $\mathcal{O}(n^3)$ time, since checking P-domination between two tuples is in $\mathcal{O}(n)$. The basic idea to reduce the actual running time is to use a 2-phase algorithm, whose general schema goes as follows. In the first phase, for each tuple u we compute the bounds $H^-(u)$ and $H^+(u)$, which requires $\mathcal{O}(n^2)$ time overall. In the second phase we actually compare tuples, and also compute the in-between probabilities, $IbP(u, v)$, for all pairs of tuples such that $u \succ v$ yet $p(u) < p(v)$.

We consider several variants of this basic schema. As a preliminary observation, it has to be remarked that the pre-processing step of topologically sorting the input relation R^p, so that a tuple u dominating v can never follow v, which is commonly used in the deterministic case [2, 4] (since it leads to a reduction of the number of comparisons and simplifies the management of the result set,

that can only grow in size), would not provide such guarantees in our scenario. This is because, as explained in Section 4, it could well be the case that $u \succ_p v$ even if $u \not\succ v$. However, as detailed below, sorting can be exploited to speed up the computation of the H^- and H^+ bounds and of the quantities $IbP(u, v)$.

The baseline algorithm for computing the bounds $H^-(u)$ and $H^+(u)$ precisely follows their definition, given in Section 4, thus tuples in R^p are sequentially accessed and compared with all already encountered tuples. The number of comparisons is thus $n(n-1)/2$. Topologically sorting R^p only slightly reduces the running time, since if v follows u in the order then we can only conclude that v is not needed to compute $H^-(u)$.

Once all bounds are computed, the second phase of the algorithm can start, in which tuples are actually compared. Algorithm 1 resembles the well-known BNL algorithm for computing the skyline of a (non-probabilistic) relation [3]. Each tuple u of R^p is compared to all the tuples v currently in the skyline: for this, the quantity $IbP(u, v)$ (or $IbP(v, u)$) is computed at line 5/6 (only if $u \succ v$, or $v \succ u$, and Rule 1 of P-domination fails). If $u \succ_p v$, then v can be dropped from $\text{SKY}(R^p)$ (line 7); otherwise, if $v \succ_p u$, then u cannot be part of the skyline (line 8) and the loop terminates.

Algorithm 1 Tuple comparison

Input: probabilistic relation R^p, each tuple u in R^p includes bounds $H^+(u)$ and $H^-(u)$
Output: $\text{SKY}(R^p)$, the skyline of R^p

1: $\text{SKY}(R^p) \leftarrow \emptyset$
2: **for all** tuples $u \in R^p$ **do**
3: $\quad insert \leftarrow$ true
4: $\quad$ **for all** tuples $v \in \text{SKY}(R^p)$ **do**
5: $\quad\quad$ **if** $u \succ v \wedge p(u) < p(v)$ **then** $IbP(u, v) \leftarrow computeIbP(R^p, u, v)$
6: $\quad\quad$ **else if** $v \succ u \wedge p(v) < p(u)$ **then** $IbP(v, u) \leftarrow computeIbP(R^p, v, u)$
7: $\quad\quad$ **if** $u \succ_p v$ **then** $\text{SKY}(R^p) \leftarrow \text{SKY}(R^p) \setminus \{v\}$
8: $\quad\quad$ **else if** $v \succ_p u$ **then** $insert \leftarrow$ false, continue (goto 9)
9: $\quad$ **if** $insert$ **then** $\text{SKY}(R^p) \leftarrow \text{SKY}(R^p) \cup \{u\}$

Again, a topological sort of R^p guarantees that the test at line 5 in Algorithm 1 is never satisfied, thus we could obtain a faster execution of the algorithm. The computation of the value $IbP(u, v)$ can be performed in a trivial way by following the definition in Section 4, i.e., by checking if any tuple t in R^p satisfies $u \succ t \succ v$. If R^p is topologically sorted, then all such tuples can only be found between u and v, i.e., if i and k, respectively, are the indices of u and v in the sorted R^p, then we need only to check those tuples t_j such that $i < j < k$. As an alternative implementation, we could also exploit a spatial index, able to efficiently solve window queries, i.e., to find all tuples included in a hyper-rectangular region of the attribute space $\mathcal{A}$. In particular, we use an R-tree [7] for retrieving all tuples in a window whose opposite vertices consist of

the coordinates of tuples v and u, respectively: $IbP(u, v)$ can then be computed by simply summing up probabilities of result tuples.[7]

6 Experimental Evaluation

In this section we experimentally analyze the efficiency of the proposed algorithms for the computation of the skyline of a probabilistic relation. For this, we synthetically generated 100,000 4-D tuples with uniformly distributed coordinates and probability. We then contrasted the performance of the algorithms described in Section 5 when varying the data dimensionality d (only the first 2-4 coordinates are used for checking domination) and/or the data cardinality n (only a fraction of the dataset is used).

As a first result, we show in Table 2 the size of $\text{SKY}(R^p)$ for different values of d and n: this demonstrates the fact that, at least for these datasets, the skyline has always a reasonable size, thus it makes sense to actually investigate the efficiency of alternative algorithms for computing it.

$d \setminus n$	20K	40K	60K	80K	100K
2	24	29	29	32	30
3	126	154	173	186	172
4	435	619	666	781	792

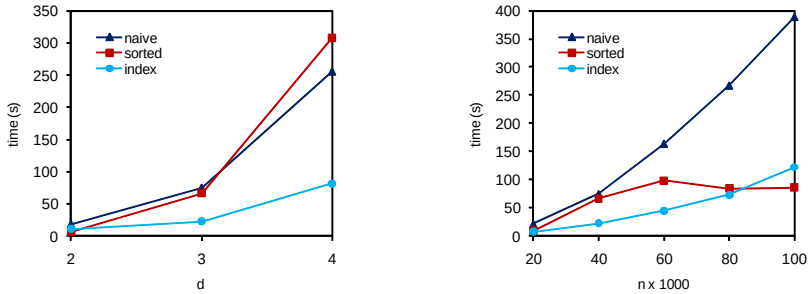
Table 2. The size of $\text{SKY}(R^p)$ for different values of d and n

In our next experiment, we evaluate the effect of topologically sorting the dataset in the first phase of the algorithm, when the H^- and the H^+ bounds are computed. As expected, sorting R^p only leads to a minor time saving: on average, the sort-based version of the algorithm is only 4% faster, with a maximum time saving of 10% (for $d = 3$ and $n = 40K$).

We then compare the performance of three variants of Algorithm 1. The variants we tested are as follows: the naive variant uses a simple loop for computing $IbP(u, v)$, sorted exploits a topological sort of R^p, so that only tuples between u and v are checked, and index uses an R-tree built on R^p. Figure 3 shows elapsed times for the three algorithms. As a first observation, we note that the index algorithm is consistently better than naive, saving around 70% of time, and that this does not depend on the data cardinality: such saving is the one provided by the index in computing the $IbP(u, v)$ values. A second, more interesting, evidence is that performance of the sorted algorithm actually improves when incrementing the dataset size: this behavior is likely due to the use of cache memory, since the comparison of consecutive tuples with a same skyline tuple requires checking almost the same sets of tuples, thus likely producing several cache hits.

Our final experiment investigates the effect of the three rules for checking P-domination between tuples. In Figure 4 (a) we show effectiveness of each rule

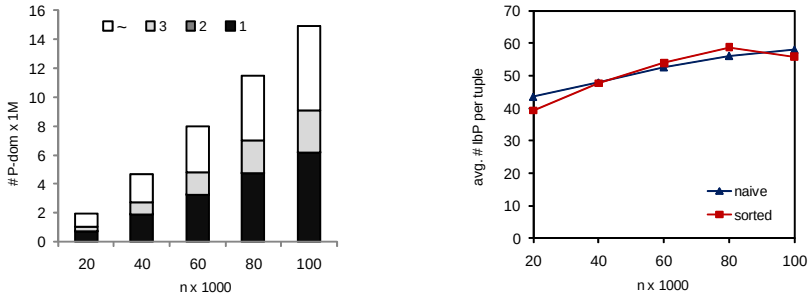
[7] Using a spatial index for the computation of the H^- and H^+ bounds would not be efficient, since it would require solving window queries with very low selectivity, unless $\mathcal{A}$ has a high dimensionality; in that case, however, the curse of dimensionality would hinder index performance.



(a) (b)

Fig. 3. Execution times for the three variants of Algorithm 1 vs. (a) dataset dimensionality ($n = 40K$) and (b) dataset cardinality ($d = 3$)

for the naive algorithm (according to our experiments, this is basically independent of the specific algorithm variant): P-domination tests linearly increase with n, as in the case of BNL-like algorithms; moreover, the most effective rule is the cheapest Rule 1 (about 40% of cases are solved with this rule), while the effectiveness of Rule 2 is less than 0.1%, so low that the graph is unable to show it. In about 40% of cases, finally, the compared tuples are indifferent. Figure 4 (b) shows the average number of $IbP(u, v)$ values that should be computed for a tuple u: clearly, this happens whenever both Rules 1 and 3 fail, as already noted in Section 5. As the figure suggests, sorting R^p has almost no effect on reducing the number of times $IbP(u, v)$ should be computed, but only, as previously observed, on the average number of tuples to be checked in each calculation.



(a) (b)

Fig. 4. Effectiveness of P-domination rules (a) and average number of IbP calculations per tuple (b) vs. dataset cardinality ($d = 3$)

7 Conclusions

In this paper we have presented a new definition of skyline for probabilistic relations, based on an appropriate definition of P-domination, i.e., domination between tuples having a confidence/probability value. We have also proved that, unlike previous definitions, ours maintains all the nice properties that skylines have in the deterministic scenario. We have provided alternative algorithms for the efficient computation of the skyline and evaluated their performance through some preliminary experiments over synthetically generated datasets.

Although we elaborated our analysis for the case of independent tuples, the definition of P-domination can be smoothly extended to the correlated case, i.e., where possible worlds are generated through a set of generation rules. This requires opportunely adapting domination rules in Section 4.1 and algorithms in Section 5, the latter maintaining the same time complexity of the independent case.

Besides a thorough experimentation with other datasets (either with different distributions of coordinates and probabilities or real ones, if available), our current and future work includes considering alternative formulations of resolution algorithms. As a matter of fact, all our algorithms share the same 2-phase structure: we expect to attain even better performance by comparing some tuples as early as possible.

References

1. Atallah, M.J., Qi, Y.: Computing all Skyline Probabilities for Uncertain Data. In: PODS 2009. pp. 279–287. Providence, RI (Jun 2009)
2. Bartolini, I., Ciaccia, P., Patella, M.: Efficient Sort-Based Skyline Evaluation. ACM TODS 33(4), 1–45 (2008)
3. Börzsönyi, S., Kossmann, D., Stocker, K.: The Skyline Operator. In: ICDE 2001. pp. 421–430. Heidelberg, Germany (Apr 2001)
4. Chomicki, J., Godfrey, P., Gryz, J., Liang, D.: Skyline with Presorting. In: ICDE 2003. Bangalore, India (Mar 2003)
5. Cormode, G., Li, F., Yi, K.: Semantics of Ranking Queries for Probabilistic Data and Expected Ranks. In: ICDE 2009. pp. 305–316. Shanghai, China (Apr 2009)
6. Fagin, R., Lotem, A., Naor, M.: Optimal Aggregation Algorithms for Middleware. In: PODS 2001. pp. 216–226. Santa Barbara, CA (May 2001)
7. Guttman, A.: R-trees: A Dynamic Index Structure for Spatial Searching. In: SIGMOD 1984. pp. 47–57. Boston, MA (Jun 1984)
8. Li, J., Saha, B., Deshpande, A.: A Unified Approach to Ranking in Probabilistic Databases. In: VLDB 2009. pp. 502–513. Lyon, France (Aug 2009)
9. Lin, X., Zhang, Y., Zhang, W., Cheema, M.A.: Stochastic Skyline Operator. In: ICDE 2009. Hannover, Germany (Apr 2011)
10. Pei, J., Jiang, B., Li, X., Yuan, Y.: Probabilistic Skylines on Uncertain Data. In: VLDB 2007. pp. 15–26. Vienna, Austria (Sep 2007)
11. Sarma, A.D., Benjelloun, O., Halevy, A.Y., Widom, J.: Working Models for Uncertain Data. In: ICDE 2006. Atlanta, GA (Apr 2006)
12. Soliman, M.A., Ilyas, I.F., Chang, K.C.C.: Top-k Query Processing in Uncertain Databases. In: ICDE 2007. pp. 896–905. Istanbul, Turkey (Apr 2007)
13. Szpilrajn, E.: Sur l'Extension de l'Ordre Partiel. Fundamenta Mathematicae 16, 386–389 (1930)
14. Zhang, W., Lin, X., Zhang, Y., Wang, W., Yu, J.X.: Probabilistic Skyline Operator over Sliding Windows. In: ICDE 2009. pp. 1060–1071. Shanghai, China (Mar 2009)
15. Zhang, X., Chomicki, J.: On the Semantics and Evaluation of Top-k Queries in Probabilistic Databases. In: DBRank 2008. pp. 556–563. Cancun, Mexico (Apr 2008)