The Skyline of a Probabilistic Relation

Ilaria Bartolini, *Member*, *IEEE*, Paolo Ciaccia, *Member*, *IEEE*, and Marco Patella

Abstract—In a deterministic relation R, tuple u dominates tuple v if u is no worse than v on all the attributes of interest, and better than v on at least one attribute. This concept is at the heart of skyline queries, that return the set of undominated tuples in R. In this paper, we extend the notion of skyline to probabilistic relations by generalizing to this context the definition of tuple domination. Our approach is parametric in the semantics for linearly ranking probabilistic tuples and, being it based on order-theoretic principles, preserves the three fundamental properties the skyline has in the deterministic case: 1) It equals the union of all top-1 results of monotone scoring functions; 2) it requires no additional parameter; and 3) it is insensitive to actual attribute scales. We then show how domination among probabilistic tuples (or *P-domination* for short) can be efficiently checked by means of a set of rules. We detail such rules for the cases in which tuples are ranked using either the "expected rank" or the "expected score" semantics, and explain how the approach can be applied to other semantics as well. Since computing the skyline of a probabilistic relation is a time-consuming task, we introduce a family of algorithms for checking P-domination rules in an optimized way. Experiments show that these algorithms can significantly reduce the actual execution times with respect to a naïve evaluation.

Index Terms—Skyline, probabilistic relation, ranking semantics

1 Introduction

Uncertain data management has recently become a very active area of research, due to the huge number of relevant applications it has. Among them, data extraction from the web [1] (due to the imprecision of the extraction tools), data integration [2] (where data in different sources are matched with a certain degree of confidence), moving objects [3], sensor networks [4], and so on.

According to a commonly adopted model, uncertain data can be represented through *probabilistic relations*, in which each tuple has also a probability to appear [5], [6], [7]. A probabilistic relation can be viewed as a compact representation of a set of *possible worlds*, each of them being a deterministic relation formed by a subset of tuples. To capture correlation among tuples, a set of *generation rules* is also present, e.g., to state that some tuples are mutually exclusive, must coexist, and so on, which limits the number and the structure of the possible worlds.

In recent years, several works have focused on extending different query types to probabilistic databases [6], [8]. In this paper we concentrate on *skyline queries* [9], whose relevance in supporting multicriteria decision analysis is well known [10], [11]. The skyline of a relation R is the set of undominated (*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 better than v on at least one attribute. The appeal of skyline comes from the observation that it 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 which are the best alternatives in a relation.

As an example, consider a traffic-monitoring application that collects data by means of a set of radars and stores them in a database, a sample of last-hour recording being shown in Fig. 1.[1] Because of radar locations, a same car cannot be detected by two distinct radars within an interval of 1 hour. This implies, for instance, that tuples t_2 and t_3 are mutually exclusive. Each radar reading has associated a Prob value, representing the overall confidence one has in the reading.

A skyline query on the Time and Speed attributes would find those readings that, at the same time, are the most recent ones and concern high-speed cars. In the deterministic case, it would be $\mathrm{Sky}(R) = \{t_1, t_2, t_4, t_5, t_{11}\}$, because all other tuples are dominated (e.g., t_1 dominates t_9). This implies, for instance, that there exists a monotone scoring function on the Time and Speed attributes that is maximized by t_1, thus making t_1 the top-1 result for such function, but this is not the case for t_9 (i.e., t_9 can never be a top-1 tuple).

In the probabilistic case, even *defining* what the skyline should be is challenging. Intuitively, one should consider not only the ranking attributes (i.e., Time and Speed), but also the confidence of each reading. This interplay between probability and ranking attributes is a well-known issue for top-k queries [12], [14], [15], but it has never been investigated for skylines. Indeed, previous approaches to the problem (see Section 2 for details) focused on the rather different issue of *group* skyline, i.e., when the generation rules partition tuples into a set of groups, each corresponding to an uncertain object, with tuples in the same group being mutually exclusive and representing possible instances of that object. In this case, one is interested in retrieving undominated objects, rather than tuples, which leads to rather different issues.

• *The authors are with DEIS, Università di Bologna, Viale del Risorgimento, 2, 40136 Bologna, Italy.*
E-mail: {i.bartolini, paolo.ciaccia, marco.patella}@unibo.it.

Manuscript received 21 Oct. 2011; revised 26 Mar. 2012; accepted 24 Apr. 2012; published online 9 May 2012.
Recommended for acceptance by S. Greco.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-10-0649.
Digital Object Identifier no. 10.1109/TKDE.2012.102.

1. This example was also considered by previous works on top-k queries [12], [13].

TID	Plate No	Radar Loc	Time	Speed	Prob
t_1	X-123	L1	10:53	90	0.2
t_2	W-246	L2	10:50	100	0.15
t_3	W-246	L3	10:40	95	0.1
t_4	Z-456	L1	10:32	110	0.1
t_5	Z-456	L2	10:30	130	0.3
t_6	H-121	L3	10:30	110	0.2
t_7	Y-324	L4	10:30	90	0.5
t_8	X-827	L4	10:20	105	0.35
t_9	X-827	L5	10:15	90	0.4
t_{10}	C-442	L5	10:10	120	0.3
t_{11}	C-442	L2	10:05	140	0.1

Fig. 1. A probabilistic relation.

	$w_t = 0.5, w_s = 0.5$		$w_t = 0.7, w_s = 0.3$	
TID	score	exp. rank	score	exp. rank
t_1	71.5	2.11	64.1	2.03
t_2	75	2.2275	65	2.1825
t_3	67.5	2.46	56.5	2.42
t_4	71	2.445	55.4	2.445
t_5	80	**1.71**	60	1.815
t_6	70	2.17	54	2.17
t_7	60	2	48	**1.625**
t_8	62.5	2.175	45.5	2.21
t_9	52.5	2.44	37.5	2.44
t_{10}	65	2.055	43	2.31
t_{11}	72.5	2.415	45.5	2.525

Fig. 2. The effect of two scoring functions ($s(t) = w_t \cdot minutes(t.\mathrm{Time}) + w_s \cdot t.\mathrm{Speed}$) on tuples in Fig. 1 and corresponding expected ranks.

In this paper, we address the problem of *defining and efficiently computing which tuples are in the skyline of a probabilistic relation*. As a first step we provide a formal definition of skyline, which is based on a generalization to the probabilistic case of the concept of tuple domination, termed *P-domination*. We prove that our extension inherits from the deterministic case all the basic skyline properties. Further, because our skyline definition works at the *tuple level*, it applies to *any* model of tuple correlation, i.e., it is not restricted to the case in which tuples are partitioned into groups.

Because P-domination is tightly related to the semantics used to linearly rank probabilistic tuples, the skyline we obtain is parametric in such semantics. Thus, whatever semantics for top-k queries one wants to adopt, our skyline definition will be always consistent with it, which is a remarkable property. For instance, if one adopts the "expected rank" semantics [15] (see Section 4.1), the skyline of the relation in Fig. 1 will consist of tuples t_5 and t_7. Thus, only t_5 and t_7 could be top-1 results, i.e., attaining the lowest expected rank, should one use *any* scoring function that is monotone in the Time and Speed attributes. This is exemplified in Fig. 2, where two such scoring functions are considered.

We also consider how to efficiently check P-domination and, consequently, to compute the skyline. From the order-theoretic definition of P-domination one should collect all the top-1 results by varying (in infinite ways) the scoring function used to linearly rank tuples. Even considering linear orders (rather than scoring functions) the problem, although finite, would have exponential complexity. We solve the problem by showing how P-domination can be checked without the need of explicitly considering neither scoring functions nor linear orders. The intuition behind our approach is that one can prove that tuple u P-dominates tuple v if u is ranked (in a probabilistic sense) better than v even for a scoring function that maximally favors v with respect to u. We demonstrate how this approach results in specific *P-domination rules* for the expected rank and expected score semantics. For lack of space, we do not detail other semantics here; the interested reader can refer to [16].

TABLE 1
Summary of Symbols

Symbol	Description
t, u, v	tuples
R	relation (set of tuples)
p	probability function
$\mathcal{G}$	set of generation rules
$R^p = (R, p, \mathcal{G})$	probabilistic relation
$\succ$	domination relation
$\geqslant$	linear order
$\mathrm{Ext}(\succ)$	set of linear orders that extend $\succ$
$\succ_p$	probabilistic domination (P-domination) relation
Ψ	probabilistic ranking function (PRF)
$\psi_{\geqslant}(u)$	value that Ψ assigns to u
$\geqslant_p = \Psi(\geqslant, R^p)$	probabilistic linear order determined by $\geqslant$ and Ψ
G, G_u	group, group of tuple u
$H_{G,\geqslant}(u)$	overall prob. of tuples not in G_u better than u in $\geqslant$
$H^-_G(u), H^+_G(u)$	lower and upper bounds on $H_{G,\geqslant}(u)$

For reducing the actual response time of skyline computation, we introduce a family of algorithms based on a common two-phase structure. The algorithms adopt a set of optimization strategies, such as presorting tuples, spatial indexing, and P-domination rule ordering. Since such strategies are orthogonal to each other, they can be combined in a modular way. Finally, we extensively evaluate our algorithms on a variety of data sets, which show the practical applicability of our approach.

Summarizing, our contributions are:

- We introduce a novel definition of domination among probabilistic tuples (P-domination), that is valid for any ranking semantics and model of tuple correlation;
- We prove that the skyline resulting from this definition satisfies all the properties that hold in the deterministic case;
- We introduce basic principles for efficiently checking P-domination, and exemplify them for the expected rank and expected score semantics;
- We provide a family of algorithms able to efficiently compute the skyline of a probabilistic relation under the above semantics, demonstrating their applicability to a range of large data sets.

The rest of the paper is organized as follows: Section 2 introduces background definitions and surveys relevant related work. Section 3 contains the statement of the problem, showing how P-domination can be defined and checked for any pair of probabilistic tuples. In Section 4, we fully develop the analysis for the expected rank and expected score semantics. In Section 5, we give algorithms for efficiently computing the skyline under the previous semantics, that are experimentally evaluated in Section 6. Section 7 concludes the paper. Relevant symbols used in the paper are given in Table 1.

2 Preliminaries

We adopt the well-known *possible worlds semantics* [8], in which a probabilistic relation R^p represents a set of standard relations, each termed a *possible world*. Compactly,

R^p is represented as a triple, $R^p = (R, p, \mathcal{G})$, where R is a relation in the standard sense, i.e., a set of tuples, also called a *deterministic* relation; p is a function that assigns to each tuple $u \in R$ a probability, $p(u) \in (0, 1]$; and $\mathcal{G}$ is a set of *generation rules* that capture correlations among tuples. A *possible world* W of R^p is a subset of tuples of R that respect all the rules in $\mathcal{G}$, with $\Pr(W)$ denoting its probability. The set of all possible worlds of R^p is denoted $\mathcal{W}$. Notice that $\mathcal{G} = \emptyset$ corresponds to the case of independent tuples, whereas if $\mathcal{G}$ only includes mutual exclusion rules that form a partition of R, then one has the widely adopted *x-relation* model [5]. In this case, each $G \in \mathcal{G}$ is also called a *group*, and consists of a set of mutually exclusive tuples (with overall probability ≤ 1). When needed, we will use G_u to denote the group of tuple u.

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 $\mathrm{Sky}_{\mathcal{A}}(R)$ or simply $\mathrm{Sky}(R)$, 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,

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

When neither $u \succ v$ nor $v \succ u$ hold, we say that u and v are *indifferent*, written $u \sim v$.

A *scoring function* $s()$ on the attributes $\mathcal{A}$ is *monotone* iff $u.A_i \geq v.A_i$ $(i = 1, \ldots, d)$ implies $s(u) \geq s(v)$, and is also *domination preserving* if $u \succ v$ implies $s(u) > s(v)$.[2] Note that, say, the $\max$ function is monotone, but not domination-preserving, e.g., $(3, 4) \succ (1, 4)$ yet $\max\{3, 4\} = \max\{1, 4\}$.

The skyline has three important properties: 1) By definition, it requires no parameter to be specified; 2) it is insensitive to attribute scales; and 3) it equals the union of top-1 results under all possible domination-preserving monotone scoring functions.[3] In the following, we always implicitly assume that monotone functions are also domination preserving.

2.1 Related Work

Over the last few years, different ranking semantics for answering top-k queries in probabilistic databases have been introduced. Soliman et al. [12] propose two distinct semantics: *U-Topk*, which returns the most likely top-k result set, considering the probabilities of the possible worlds of R^p; and *U-kRanks*, that, for each $i = 1, \ldots, k$, returns the tuple with the highest probability of being at rank (position) i. Yi et al. [18] provide efficient algorithms for such semantics under the x-relation model of tuple correlation. Zhang and Chomicki [14] introduce the *Global-Topk* semantics, which computes for each tuple u the probability that u is among the top-k tuples in the possible worlds of R^p, and returns the k tuples with the highest probabilities. Based on the observation that none of these semantics defines a ranking of tuples in the strict sense, i.e., the top-k tuples need not to be a proper subset of the top-$(k+1)$ ones (since the actual ranking might depend on the value of k), in [15] Cormode et al. proposed the *expected rank* semantics. Here, one computes the rank that tuple u has in each of the possible worlds of R^p, and then takes the expected value of such ranks. The *expected score* semantics [15] ranks tuples by simply taking the product of score and probability. A discussion on the properties of the various ranking semantics can be found in [14], [15], [13].

In [13], Li et al. argue that no top-k semantics is suitable for all possible practical cases, rather the most appropriate choice might depend on the data set at hand. Thus, they propose a unifying framework for ranking based on a parameterized ranking function $\Upsilon_\omega()$, and show how this can be used to approximate specific ranking methods. Remarkably, the correlation model in [13] (*and/xor trees*) also allows for coexistence rules; thus, it is far more general than the x-relation (groups) model usually considered for top-k ranking algorithms.

The first work to consider skyline queries on probabilistic data has been [19]. There, the basic idea is to compute for each tuple u the probability that u is undominated, and then rank tuples based on these *skyline probabilities*. The skyline probability of u, $\Pr_{\mathrm{Sky}}(u)$, is the overall probability of the possible worlds W in which u is in the skyline of W, which for the x-relation model (the one assumed by Pei et al. [19]) is derived to be:

$$\Pr_{\mathrm{Sky}}(u) = p(u) \times \prod_{G \neq G_u} \left(1 - \sum_{t \in G, t \succ u} p(t)\right). \tag{2}$$

Finally, the p-skyline of R^p, based on a probability threshold p, is defined as the set of tuples u having a skyline probability not lower than p, $\Pr_{\mathrm{Sky}}(u) \geq p$.

It has to be observed that this approach is unable to preserve the basic skyline properties, because 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 [20] and shown how to compute p-skylines on uncertain data streams [21].

More recently, Lin et al. have proposed the *stochastic skyline operator* [22]. Given two uncertain objects (i.e., groups) U and V, each characterized by a discrete probability distribution, U stochastically dominates V iff, for each point $\mathbf{x}$ in the attribute space, the cumulative probability of U in $\mathbf{x}$, $U.cdf(\mathbf{x})$, is at least equal to $V.cdf(\mathbf{x})$, and there exists a point $\mathbf{y}$ such that $U.cdf(\mathbf{y}) > V.cdf(\mathbf{y})$. This method, although not requiring any additional parameter, has some major limitations. First, testing stochastic domination is an NP-complete problem; thus, the method does not scale well with the number of skyline attributes. Second, the stochastic skyline equals only a *subset* of possible top-1 results, namely those arising from the *expectation of multiplicative scoring functions*, i.e., $f(U) = E[\prod_{i=1}^{d} f_i(A_i)]$. Finally, by definition it is only applicable to the x-relation model, because it works at the group level.

2. Domination-preserving monotone functions coincide with those that others call *strictly monotone in each argument* [17].

3. Although it is folklore that the skyline of R equals the union of top-1 results under all possible monotone scoring functions, this is imprecise because of the non-deterministic nature of top-1 queries. For instance, given $R = \{(3, 4), (1, 4)\}$ and the $\max$ function, $(1, 4)$ might be (nondeterministically) returned as the top-1 result.

3 P-Domination and the Skyline of a Probabilistic Relation

In this section, we provide a definition of skyline for probabilistic relations that is applicable to any ranking semantics and model of tuple correlation, and prove that it preserves all the basic properties the skyline has in the deterministic case. We start by rewriting (1) as:

$$\boxed{\mathrm{Sky}(R^p) = \{u \in R \mid \nexists\, v \in R : v \succ_p u\ \}} \tag{3}$$

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

To preserve all skyline properties, we define P-domination using principles of order theory. In order-theoretic terms, $\succ$ is a *strict partial order*, i.e., an irreflexive ($\forall u : u \not\succ u$) and transitive ($\forall u, v, t : u \succ v \wedge v \succ t \Rightarrow u \succ t$) binary relation. 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 $v \succcurlyeq u$.[4] A linear order $\succcurlyeq$ is a *linear extension* of $\succ$ iff $u \succ v \Rightarrow u \succcurlyeq v$, that is, $\succcurlyeq$ is *compatible* with $\succ$. Notice that a linear extension $\succcurlyeq$ of $\succ$ is what one obtains by ordering tuples with a monotone scoring function $s()$ and then breaking ties arbitrarily.

Let $\mathrm{Ext}(\succ)$ denote the set of all linear extensions of $\succ$. A fundamental result in order theory, derived from *Szpilrajn's Theorem* [23], asserts that *any strict partial order $\succ$ equals the intersection of its linear extensions*, i.e.,

$$\succ = \bigcap \{\succcurlyeq \mid \succcurlyeq \in \mathrm{Ext}(\succ)\}.$$

This is the first ingredient needed to define P-domination.

Our second ingredient exploits the observation that *a 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 is what semantics for top-k queries that define a ranking in the proper sense of the term, such as expected rank and expected score, do, and, as discussed below, also applies to other semantics by considering for them the ranking induced on tuples when $k = 1$. Next definition formalizes this concept.

Definition 1 (Probabilistic Ranking Function (PRF)). *Let $R^p = (R, p, \mathcal{G})$ be a probabilistic relation, and let $\succcurlyeq$ be a linear ordering of the tuples of R. A PRF Ψ is a function that takes as input R^p and $\succcurlyeq$ and yields a (new) linear ordering of the probabilistic tuples of R^p. The resulting order is called a* probabilistic linear order, *denoted $\succcurlyeq_p \stackrel{\mathrm{def}}{=} \Psi(\succcurlyeq, R^p)$. The actual ranking of the probabilistic tuples is obtained by computing for each tuple u a value $\psi_{\succcurlyeq}(u)$, so that $u \succcurlyeq_p v$ iff $\psi_{\succcurlyeq}(u) > \psi_{\succcurlyeq}(v)$.*

Notice that, although in practice one might want to use a PRF that has some desirable and intuitive properties, such as those discussed in [14], [15], [13], in the following we do not put any restriction on the specific Ψ used to rank tuples, so as to gain in generality. We are now ready to define P-domination:

4. 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.

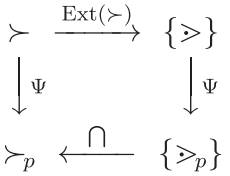

Fig. 3. How P-domination is defined.

Definition 2 (P-Domination). *Let $R^p = (R, p, \mathcal{G})$ 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 PRF 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 $\succcurlyeq_p = \Psi(\succcurlyeq, R^p)$, it is $u \succcurlyeq_p v$, that is:*

$$\boxed{u \succ_p v \Leftrightarrow u \succcurlyeq_p v,\ \ \forall\ \succcurlyeq_p = \Psi(\succcurlyeq, R^p), \succcurlyeq \in \mathrm{Ext}(\succ)} \tag{4}$$

The diagram in Fig. 3 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. The arrow on the left adorned with "Ψ" emphasizes that P-domination is parametric in the semantics adopted for linearly ranking probabilistic tuples.

The following result follows from Definition 2:

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

Proof. Clearly, $\succ_p$ is irreflexive. To prove that also transitivity holds, assume that $u \succ_p v$ and $v \succ_p t$ both hold. Given Ψ, this implies that, for each linear extension $\succcurlyeq$ of $\succ$ and associated probabilistic linear order $\succcurlyeq_p = \Psi(\succcurlyeq, R^p)$, it is $u \succcurlyeq_p v$ and $v \succcurlyeq_p t$, thus $u \succcurlyeq_p t$ as required. □

Next theorem extends to the probabilistic realm the fundamental property that the skyline contains all and only the top-1 results of (domination preserving) monotone scoring functions.

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

Proof. (if) Assume that for some scoring function $s()$, with induced linear order $\succcurlyeq$, tuple $u \in R^p$ is the top-1 for the probabilistic order $\succcurlyeq_p = \Psi(\succcurlyeq, R^p)$. To prove that $u \in \mathrm{Sky}(R^p)$, we have to show that no tuple v P-dominates u. By contradiction, assume this is not the case, i.e., there exists v such that $v \succ_p u$. In turn, this implies $v \succcurlyeq_p u$ for all probabilistic rankings obtained from a linear extension of $\succ$, thus u can never be a top-1 tuple, a contradiction.

(only if) We prove that if $u \in \mathrm{Sky}(R^p)$, then there exists $s()$, with induced linear order $\succcurlyeq$, such that u is the top-1 for the probabilistic order $\succcurlyeq_p = \Psi(\succcurlyeq, R^p)$. By hypothesis, for any tuple v either u P-dominates v or

the two are probabilistically indifferent, written $u \sim_p v$. To prove the result we make use of a lemma by Szpilrajn [23], that asserts that if two tuples are indifferent, $u \sim_p v$, then there exist a strict partial order $\succ'_p$ that includes $\succ_p$ and such that $u \succ'_p v$. By applying the lemma to all the tuples v that are probabilistically indifferent to u (and arbitrarily ordering all other pairs of probabilistically indifferent tuples), it is, therefore, possible to derive a probabilistic linear order $\succcurlyeq_p$ in which u is the top-1 tuple, from which the result follows. □

Further, as in the deterministic case, the skyline of R^p is insensitive to actual attribute values, rather it only depends on the relative ordering on each skyline attribute.

Theorem 3. *Let $R^p = (R, p, \mathcal{G})$ be a probabilistic relation, and let $\mathcal{A}$ be the skyline attributes. Let $S^p = (S, p, \mathcal{G})$ be a probabilistic relation obtained from R^p as follows: S is obtained from R through an isomorphism ϕ that preserves Pareto domination, i.e., $\forall u, v \in R$ it is $\phi(u) \succ \phi(v)$ iff $u \succ v$; $\forall u \in R$ it is $p(\phi(u)) = p(u)$ (probabilities are preserved); $\forall u \in R$, $\forall G \in \mathcal{G}$, u in G is replaced by $\phi(u)$ in G (rules do not change as well). Then, for any PRF Ψ, it is $\mathrm{Sky}(R^p) = \mathrm{Sky}(S^p)$.*

Proof. Because of the theorem's hypotheses and of the observation that actual attribute values are not used at all to define P-domination (and consequently the skyline of R^p), the result immediately follows. □

Due to Theorems 2 and 3, and to the fact that no parameter at all is used, it is proved that our skyline definition benefits all the three properties the skyline has in the deterministic case. We want to stress that all the results presented so far apply regardless of the specific PRF used as well as of the model of tuple correlation, which only impact on the complexity of algorithms for computing the skyline, but not on its definition and properties.

We conclude this section by considering two issues that are of practical interest: 1) How P-domination can be defined for those semantics for which the ranking of tuples varies with the value of k; and 2) how to deal with PRF's that admit ties.

As to issue 1, the solution is indeed simple and consists of considering the case of top-1 queries. This holds since the skyline is the union of top-1 answers, thus we can limit the analysis to the case $k = 1$, and report as $\mathrm{Sky}(R^p)$ those tuples that, for at least one linear order $\succcurlyeq$ compatible with $\succ$, are the top-ranked ones. More precisely, let $\psi^{(1)}_{\succcurlyeq}(u)$ be the value that a ranking semantics for top-1 queries, denoted $\Psi^{(1)}$, assigns to tuple u for computing the result of a (top-1) query on R^p. Then, it is plain to see that if one considers $\Psi^{(1)}$ and $\psi^{(1)}_{\succcurlyeq}()$ in place of Ψ and $\psi_{\succcurlyeq}()$, respectively, all the results we have derived still apply.

When the PRF admits ties, i.e., for some tuples u and v it is $\psi_{\succcurlyeq}(u) = \psi_{\succcurlyeq}(v)$, we distinguish two cases. If $u \succ v$, then we insist to apply a domination-preserving tie-breaking rule, so that $u \succcurlyeq_p v$. On the other hand, if $u \sim v$, then we stipulate that it has to be $u \succ_p v$ only if no (other) probabilistic linear order exists such that $\psi_{\succcurlyeq}(u) < \psi_{\succcurlyeq}(v)$ holds. The rationale behind this choice is to keep the analogy with the deterministic case, in which $u \succ v$ if it is never the case that $s(u) < s(v)$ should one consider *all* monotone, i.e., not necessarily domination-preserving, functions (in which case it can well be $s(u) = s(v)$ even if $u \succ v$). Notice that having formulated our definitions and results (e.g., Definition 2 and Theorem 2) so as to explicitly consider also PRF's admitting ties would have been possible, yet made unnecessarily more complex the presentation.

3.1 How to Check P-Domination

Definition 2 cannot be directly used to check P-domination, because it requires to enumerate all linear extensions of $\succ$, and these can be exponential in the number of tuples.[5] The approach we pursue, and that can be applied regardless of the specific ranking semantics Ψ, does not require to materialize *any* linear extension of $\succ$ and, therefore, results in more efficient algorithms for skyline computation.

According to Definition 2, it is $u \succ_p v$ iff $\psi_{\succcurlyeq}(u) > \psi_{\succcurlyeq}(v)$ holds for all linear extensions $\succcurlyeq$ of $\succ$, i.e.,

$$\min_{\succcurlyeq \in \mathrm{Ext}(\succ)} \left\{ \frac{\psi_{\succcurlyeq}(u)}{\psi_{\succcurlyeq}(v)} \right\} > 1. \tag{5}$$

The key idea for efficiently checking the above inequality without enumerating the linear extensions of $\succ$ is to determine which is the one 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 PRF Ψ, there are two cases to consider:

$\mathbf{u \succ v}$: When u dominates v, we can restrict the analysis to those linear orders for which it is $u \succcurlyeq v$; among them, we determine how other tuples t that are indifferent to either u and v (or both) should be arranged in the linear order so as to minimize the ratio $\psi_{\succcurlyeq}(u)/\psi_{\succcurlyeq}(v)$.

$\mathbf{u \not\succ v}$: 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.

Summarizing, this approach will result in a set of *P-domination rules* that can be checked without the need of materializing any linear extension of $\succ$.

4 P-Domination Rules

Unlike the general concepts and results introduced so far, P-domination rules depend on the PRF Ψ used, as well as on the model of tuple correlation. Since an exhaustive enumeration of all known cases would require too much space, in the following we only consider the expected rank and the expected score semantics under the x-relation model (as in [15]), and then briefly discuss other semantics. Notice that this choice does not imply any bias on our side toward the expected rank and expected score semantics, rather we just found them better suited to illustrate how P-domination rules are derived, because such semantics define a ranking in the proper sense of the term.

4.1 P-Domination with Expected Ranks

In [15], 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

5. If R^p consists of N pairwise indifferent tuples, then $\mathrm{Ext}(\succ)$ has size $N!$, since each permutation is compatible with $\succ$.

W with $|W|$ tuples is the number of tuples in W that precede u, i.e.,

$$\text{rank}_{W,\succ}(u) = \begin{cases} |\{v \in W | v \succ u\}| & \text{if } u \in W \\ |W| & \text{otherwise.} \end{cases}$$

Therefore, the expected rank of a tuple u is

$$ER_{\succ}(u) = \sum_{W \in \mathcal{W}} \text{rank}_{W,\succ}(u) \times \Pr(W).$$

As in [15], we consider that in case two tuples have a same expected rank value, a (domination-preserving) tie-breaking rule is applied so that expected ranks define a linear order. Let $\succ_p$ be such linear order, i.e., $u \succ_p v$ iff $ER_{\succ}(u) < ER_{\succ}(v)$. Thus, for this semantics one could set $\psi_{\succ}(u) = -ER_{\succ}(u)$.

Following [15], the expected rank of a tuple u is

$$ER_{\succ}(u) = p(u) \times \sum_{t \succ u, t \notin G_u} p(t) + (1 - p(u)) \times \sum_{t \notin G_u} p(t) + \sum_{t \in G_u, t \neq u} p(t),$$

where the first term is the expected rank of u in a possible world in which u appears, whereas the other terms account for the contribution of those worlds that do not contain u.

Let $H_{G,\succ}(u) = \sum_{t \succ u, t \notin G_u} p(t)$ be the overall probability of those tuples not in G_u that are better than u according to $\succ$. Further, let P be the overall probability of all the tuples in R^p, $P = \sum_{t \in R^p} p(t)$, and $P_{G_u} = \sum_{t \in G_u, t \neq u} p(t)$. Then, $ER_{\succ}(u)$ can also be compactly written as:

$$ER_{\succ}(u) = p(u) \times H_{G,\succ}(u) + (1 - p(u)) \times (P - P_{G_u} - p(u)) + P_{G_u}.$$

According to Definition 2, tuple u P-dominates v iff it is $\max_{\succ \in \text{Ext}(\succ)}\{ER_{\succ}(u)/ER_{\succ}(v)\} < 1$. After defining $P_{u,v} = P - p(u) - p(v)$, this inequality can be equivalently written as (see [16] for the detailed algebraic steps):

$$u \succ_p v \Leftrightarrow \boxed{\frac{p(u)}{p(v)} > \max_{\succ \in \text{Ext}(\succ)} \left\{ \frac{P_{u,v} + 1 - H_{G,\succ}(v) - P_{G_v}}{P_{u,v} + 1 - H_{G,\succ}(u) - P_{G_u}} \right\}} \quad (6)$$

The two cases to consider for checking the validity of Inequality 6 are dealt with as follows: For both here we describe the general case in which u and v belong to different groups first, postponing the (easier) case $G_u = G_v$.

$\mathbf{u \succ v}$: The first rule we derive is sufficient (but not necessary) for P-domination to occur and has the advantage that it can be more efficiently checked than the other we derive for the case $u \succ v$.

Since $u \succ v$, it is $H_{G,\succ}(v) \geq H_{G,\succ}(u) + p(u) - P_{G_v}$, which is obtained by assuming a worst-case scenario for u in which all tuples $t \in G_v$ other than v dominate u. Substituting in Inequality 6 it is obtained:

$$\boxed{u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq 1 \ \wedge \ \frac{p(u)}{P_{G_u}} \geq 1} \quad \text{(ER:Rule 1)}$$

Note that in case neither inequality holds strictly it is still safe to say that $u \succ_p v$, because we assume a domination-preserving tie-breaking rule.

In case Rule 1 is not satisfied, a preliminary key observation is that, for any linear order $\succ$ that extends $\succ$, it is $H_{G,\succ}(u) \in [H^-_G(u), H^+_G(u)]$, where the two bounds are, respectively, defined as:

$$H^-_G(u) = \sum_{t \succ u, t \notin G_u} p(t) \qquad H^+_G(u) = \sum_{u \not\succ t, t \notin G_u} p(t).$$

Clearly, $H^-_G(u)$ is the best possible case for u, because only those tuples from other groups that dominate u are also better than u according to $\succ$, whereas the worst case for u arises when u is better only of those tuples of other groups that it dominates.

The ratio in the right-hand side of Inequality 6 can be maximized by adequately arranging those tuples t that are indifferent to either u or v (or both). Since $u \succ v$ and $\succ$ is transitive, only three cases can occur:

$\mathbf{t \sim u, t \succ v}$: To favor v with respect to u, it has necessarily to be $t \succ u$ whenever $t \notin G_u$. Let $IB(u,v) = \sum_{t \sim u, t \succ v, t \notin G_u} p(t)$ stand for the mass of probability of such tuples.

$\mathbf{t \sim v, u \succ t}$: For the same reason it has to be $v \succ t$.

$\mathbf{t \sim u, t \sim v}$: To analyze this case, first observe that setting $t \succ u \succ v$ when $t \in G_v$ would penalize only u. Let $II_{G_v}(u,v) = \sum_{t \sim u, t \sim v, t \in G_v} p(t)$ be the total mass of probability of such tuples. At this point the ratio in Inequality 6 would be written as:

$$\frac{N(u,v)}{D(u,v)} \stackrel{\text{def}}{=} \frac{P_{u,v} + 1 - H^-_G(v) - P_{G_v}}{P_{u,v} + 1 - H^-_G(u) - IB(u,v) - II_{G_v}(u,v) - P_{G_u}}. \quad (7)$$

Consider now those tuples $t \notin G_u \cup G_v$, whose total mass of probability is

$$\Delta(u,v) = \sum_{t \sim u, t \sim v, t \notin G_u, t \notin G_v} p(t).$$

The two alternatives to consider are either $t \succ u \succ v$ or $u \succ v \succ t$. The choice actually depends on the value of $N(u,v)/D(u,v)$. If $N(u,v) < D(u,v)$, then setting $t \succ u \succ v$ would lead to the ratio $(N(u,v) - \Delta(u,v)) \ / \ (D(u,v) - \Delta(u,v))$, which is *lower* than $N(u,v) \ / \ D(u,v)$. On the other hand, when $N(u,v) > D(u,v)$, then $(N(u,v) - \Delta(u,v)) \ / \ (D(u,v) - \Delta(u,v)) > N(u,v)/D(u,v)$, thus $t \succ u \succ v$ is the case to consider. Finally, when $N(u,v) = D(u,v)$ Inequality 6 degenerates to $p(u) > p(v)$.

Putting all together, the second P-domination rule is:

$$\boxed{u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq \begin{cases} \dfrac{N(u,v)}{D(u,v)} & \text{if } \dfrac{N(u,v)}{D(u,v)} \leq 1 \\ \dfrac{N(u,v) - \Delta(u,v)}{D(u,v) - \Delta(u,v)} & \text{otherwise} \end{cases}} \quad \text{(ER:Rule 2)}$$

where equality is again due to the domination-preserving rule for breaking ties.

$\mathbf{u \not\succ v}$: When u does not dominate v, P-domination can still occur. In this case, it is immediate to see that the ratio in Inequality 6 is maximized by setting $H_{G,\succ}(v) = H^-_G(v)$ and $H_{G,\succ}(u) = H^+_G(u)$, thus:

TABLE 2
Probabilities and Bounds for the Data Set in Fig. 1

tuple	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
p	0.2	0.15	0.1	0.1	0.3	0.2	0.5	0.35	0.4	0.3	0.1
H_G^-	0	0	0	0	0	0.4	1.05	0.6	1.55	0.3	0
H_G^+	1.6	1.55	1.55	0.85	0.55	1.25	1.8	1.95	1.95	2.3	2.3
$\Pr_{\text{Sky}}$	0.2	0.15	0.1	0.1	0.3	0.12	0.14	0.17	0.06	0.21	0.1

$$u \not\succ v \ \wedge \ \frac{p(u)}{p(v)} > \frac{P_{u,v} + 1 - H_G^-(v) - P_{G_v}}{P_{u,v} + 1 - H_G^+(u) - P_{G_u}} \qquad \text{(ER:Rule 3)}$$

We conclude the analysis by detailing P-domination rules for mutually exclusive tuples, i.e., tuples belonging to a same group G.

1. Rule 1: Since $u \succ v$, now it is $H_{G,\gtrdot}(v) \geq H_{G,\gtrdot}(u)$; thus, the first P-domination rule reduces to:

$$u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq 1 \qquad \text{(ER:Rule 1-sg)}$$

   Note that when $p(u) = p(v)$, u and v would have the same expected rank, in which case $u \succ_p v$ holds by the domination-preserving tie-breaking rule.
2. Rule 2: The only difference with the scenario analyzed in Section 4.1 is that, since the quantity $II_{G_v}(u,v) = \sum_{t \sim u, t \sim v, t \in G_v} p(t)$ now does not contribute to $H_{G,\gtrdot}(u)$, this term vanishes from the denominator in (7). Apart from this the analysis proceeds as in the case of different groups.
3. Rule 3: This rule applies unchanged.

Example 1. Table 2 lists the probabilities of the tuples in Fig. 1, whose overall probability is $P = 2.7$, together with their H_G^- and H_G^+ bounds.

As an example of how bounds are computed consider tuple t_8. Since t_8 is dominated only by t_4, t_5, and t_6, it is $H_G^-(t_8) = p(t_4) + p(t_5) + p(t_6) = 0.6$. The only tuple dominated by t_8 is t_9, thus $H_G^+(t_8) = P - p(t_8) - p(t_9) = 1.95$.

A case to which Rule 1 applies concerns tuples t_5 and t_6, since $t_5 \succ t_6$, $0.3 = p(t_5) \geq p(t_6) = 0.2$, and $0.3 = p(t_5) \geq P_{G_{t_5}} = 0.1$. Rule 1 allows tuples t_3, t_6, t_9, and t_{10} to be immediately discarded, since they are P-dominated by t_2, t_5, t_7, and t_5 again, respectively. Rule 2 is used to show that $t_5 \succ_p t_8$ and $t_5 \succ_p t_9$. A case to which Rule 3 applies is to prove that $t_7 \succ_p t_2$ holds. Notice that not only $t_7 \not\succ t_2$, but $t_2 \succ t_7$ holds in the deterministic case. An exhaustive analysis shows that the skyline is $\text{Sky}(R^p) = \{t_5, t_7\}$.

The last row of Table 2 shows the (rounded) skyline probabilities, $\Pr_{\text{Sky}}$, of all tuples (see also (2) in Section 2.1). Although t_5, which belongs to both $\text{Sky}(R)$ and $\text{Sky}(R^p)$, has also the highest skyline probability, this is not the case for tuple t_7, which would be ranked only sixth according to the method of [19]. In general, there is no clear correlation among P-domination and $\Pr_{\text{Sky}}(u)$, i.e., a tuple might have a high skyline probability and at the same time to be P-dominated by several other tuples.

4.2 P-Domination with Expected Scores

In [15], the case where tuples are ranked by their *expected score*, $ES(u) = s(u) \times p(u)$, is also considered. For this PRF, it is immediate to derive that:

$$u \succ v \ \wedge \ \frac{p(u)}{p(v)} \geq 1 \qquad \text{(ES:Rule 1)}$$

This holds since, for any scoring function $s()$, it has to be $s(u) \times p(u) > s(v) \times p(v)$, i.e., $p(u)/p(v) > s(v)/s(u)$. If $u \not\succ v$, then for any value of $p(u)/p(v)$ the right-hand side can be made arbitrarily large and P-domination does not hold. If $u \succ v$, it is $s(v)/s(u) = 1 - \epsilon$, with $\epsilon > 0$ arbitrarily small, from which the result follows.

It is remarkable that, although ranking by expected scores is highly dependent on actual score values, the skyline of R^p is, even in this case, insensitive to attribute values. A major pitfall of expected score ranking is that it ignores correlation among tuples, and this obviously propagates to skylines. Further, because the above is the *only* P-domination rule, the skyline obtained from the expected score semantics will be always larger than the skyline in the deterministic case (since $u \succ v$ is a necessary condition).

4.3 Other Ranking Semantics

For the U-Topk, U-kRanks, and Global-Topk semantics, we can provide a unified analysis, due to the following preliminary observation (see also [24]):

Observation 1. For any probabilistic relation R^p, whose tuples are linearly ordered according to $\gtrdot$, the result of a top-1 query computed according to the U-Top1, U-1Ranks, and Global-Top1 semantics is the same, and is given by the tuple u that maximizes the top-1 probability, which for x-relations is:

$$\Pr_{\gtrdot}^{top1}(u) = p(u) \times \prod_{G \neq G_u} \left(1 - \sum_{t \in G, t \gtrdot u} p(t)\right). \qquad (8)$$

Thus, as explained at the end of Section 3, for such semantics it is $\psi_{\gtrdot}(u) = \psi_{\gtrdot}^{(1)}(u) = \Pr_{\gtrdot}^{top1}(u)$. Now, substituting in (5), and setting $Q_{G,\gtrdot}(u) = \prod_{G \neq G_u}(1 - \sum_{t \gtrdot u, t \in G} p(t))$, it is obtained:

$$u \succ_p v \Leftrightarrow \boxed{\frac{p(u)}{p(v)} > \max_{\gtrdot \in \text{Ext}(\succ)} \left\{ \frac{Q_{G,\gtrdot}(v)}{Q_{G,\gtrdot}(u)} \right\}} \qquad (9)$$

For deriving P-domination rules for this scenario, one can then follow steps similar to those applied with the expected rank semantics, as detailed in [16]. Obviously, for more complex correlation models (8) and (9) no longer apply, yet ranking tuples using their top-1 probability still remains valid.

5 Algorithms

In this section, we introduce efficient algorithms for computing the skyline of a probabilistic relation R^p consisting of N tuples.

We first cover the simpler case of the expected scores semantics. In this scenario, if we consider the probability of a tuple as a new attribute, the P-domination rule reported in Section 4.2 is equivalent to the definition of (deterministic)

TABLE 3
Alternatives Available for the Two-Phase Algorithm

Phase	symbol	description
I.a	unsrt	no sort
	srt	uses topological sort
I.b	$\emptyset$	no P-domination rule used
	R1	uses Rule 1 only
	R1R3$^+$	uses Rules 1 and 3$^+$
II.a	noidx	no index
	idx	uses a spatial index
II.b	1scan	single scan using all Rules
	2scans	1st scan: Rules 1 and 3; 2nd scan: Rule 2

domination on the $\mathcal{A} \cup \{p\}$ attributes. Consequently, *any* deterministic skyline algorithm (like BNL [9], SFS [25], or SaLSA [26]) that considers the set of attributes $\mathcal{A} \cup \{p\}$ can be used for computing $\mathrm{Sky}(R^p)$. Therefore, it is immediately derived that the skyline of R^p can be computed in $\mathcal{O}(N^2)$ time.[6]

Turning to consider the expected rank semantics, it is important to first characterize the time complexity of the three P-domination rules described in Section 4.1:

1. Rule 1: In the worst case, the evaluation over all pairs of tuples requires $\mathcal{O}(N^2)$ time. This clearly holds when the probability of each group is known in advance, but it remains true even if group probabilities need to be determined (since this can be done before starting to compare tuples, see Section 5.1).
2. Rule 3: A single check on a pair of tuples apparently has a complexity of $\mathcal{O}(N)$, because of $H_G^-(v)$ and $H_G^+(u)$; however, such values do not depend on the specific pair of tuples being compared, and thus can be precomputed for each tuple, which collectively requires $\mathcal{O}(N^2)$ time. Thus, a single evaluation is a constant time, $\mathcal{O}(1)$, operation and evaluation on all pairs of tuples requires $\mathcal{O}(N^2)$ time.
3. Rule 2: A single evaluation requires $\mathcal{O}(N)$ time, because of $IB(u,v)$, $II_{G_v}(u,v)$, and, possibly, $\Delta(u,v)$. Unless one can afford a $\mathcal{O}(N^2)$ storage overhead for book-keeping (which we do not consider here), such quantities cannot be computed in advance because they depend on both u and v.

Therefore, because of Rule 2, computing the skyline of R^p requires $\mathcal{O}(N^3)$ time in the worst case.

Based on above observations, our algorithms are designed around a common 2-phase template, shown in Algorithm 1:

Algorithm 1. 2-phase Algorithm template
Input: probabilistic relation R^p
Output: $\mathrm{Sky}(R^p)$, the skyline of R^p
PHASE I (bounds computation and low-cost pruning)
I.a) *determine whether tuples have to be sorted or not; compute basic groups' information*
I.b) *precompute $H_G^-(u)$ and $H_G^+(u)$ for each tuple u, possibly applying some P-domination rule*

6. When the data set fits in main memory, [27] provides near-linear time algorithms, whereas [28] gives subquadratic worst-case bounds for I/O complexity.

Plate No (hash key)	$\sum_{t \in G} p(t)$	tuples
X-123	0.2	$\{t_1\}$
W-246	0.25	$\{t_2, t_3\}$
Z-456	0.4	$\{t_4, t_5\}$
H-121	0.2	$\{t_6\}$
Y-324	0.5	$\{t_7\}$
X-827	0.75	$\{t_8, t_9\}$
C-442	0.4	$\{t_{10}, t_{11}\}$

Fig. 4. `groupsHT` for the relation in Fig. 1.

PHASE II (skyline computation)
II.a) *determine whether a spatial index has to be built or not*
II.b) *determine how to apply the remaining P-domination rules*

The common rationale of our algorithms is to first precompute, for each tuple u, the bounds $H_G^-(u)$ and $H_G^+(u)$ (Phase I.b). While doing this, one can also consider to apply Rule 1 to immediately discard some tuples, thus avoiding to compute their bounds. Presorting tuples (Phase I.a) can turn to be beneficial both for speeding up bounds computation, and for allowing a modified version of Rule 3 to be also applied (see Section 5.1).

In the second phase, we actually compute the skyline of R^p. Due to the high cost of evaluating Rule 2, here we consider the possibility of a 2-scan approach (Phase II.b), the first of which only involves Rule 1 (if not already applied) and Rule 3. To allow for efficiently testing Rule 2, in Phase II.a we also consider the possibility of building on-the-fly a spatial index.

Table 3 summarizes the options available for the two phases, together with symbols that will be used in graphs in Section 6.

5.1 Phase I

The main purpose of Phase I is to compute the H_G^- and H_G^+ bounds. The basic case to consider is the one where no particular order is imposed on tuples in R^p. In this case, the first action performed by Phase I.a is to build an hash table, called `groupsHT`. Each entry of `groupsHT` corresponds to a specific group G, containing (backward) links to all tuples in G and the sum of their probabilities, i.e., $\sum_{t \in G} p(t)$. Assuming direct access to `groupsHT` is in $\mathcal{O}(1)$, computing the probabilities of all groups requires $\mathcal{O}(N)$ time. Besides being used whenever group probabilities are needed to check P-domination rules, `groupsHT` is the key for the efficient implementation of Rule 2, as detailed in Section 5.2. Fig. 4 shows how `groupsHT` looks like for the data shown in Fig. 1.

In the unordered scenario, a naïve nested loops is the simplest way to compute the bounds. This basic schema can be improved if, when computing $H_G^-(v)$ and $H_G^+(v)$ for a tuple v, we halt as soon as we find a tuple u that P-dominates v by using Rule 1.

An alternative implementation of Phase I.a is to perform a topological sort of the tuples in R^p so that, if $u \succ v$, then u will precede v in the resulting order. Sorting tuples is executed before building the `groupsHT` hash table (so as to avoid updating backward links). Presorting tuples has several benefits:

- $H_G^-(u)$ can be computed by just looking at tuples t that *precede* u.

- Similarly, the computation of $H_G^+(u)$ needs to look only at tuples t that *follow* u. To see why this is the case, observe that $H_G^+(u)$ can be equivalently expressed as

$$H_G^+(u) = P - (P_{G_u} + p(u)) - \sum_{u \succ t, t \notin G_u} p(t),$$

where P is the overall probability of all tuples in R^p and $P_{G_u} + p(u)$ is the probability of all tuples in G_u. Since $u \succ t$ implies that t follows u, the assert holds.
- It is also evident that Rule 1 needs to be checked to determine if $u \succ_p v$ only if u precedes v.
- Finally, a topological sort of tuples allows for an approximate version of Rule 3 (Rule 3^+) to be applied. Let us define $\widehat{H_G^+}(u; v)$ as the running value of $H_G^+(u)$ at the time v is accessed (this value will equal $H_G^+(u)$ when all tuples have been scanned), i.e.,

$$\widehat{H_G^+}(u; v) = P - (P_{G_u} + p(u)) - \sum_{\substack{u \succ t, t \notin G_u \\ t \text{ precedes } v}} p(t).$$

Notice that the sum equals 0 if v precedes u. Since $\widehat{H_G^+}(u; v) \geq H_G^+(u)$, then it is possible to bound from above the ratio in the right-hand side of Rule 3, which leads to:

$$u \not\succ v \ \wedge \ \frac{p(u)}{p(v)} > \frac{P_{u,v} + 1 - H_G^-(v) - P_{G_v}}{P_{u,v} + 1 - \widehat{H_G^+}(u; v) - P_{G_u}} \quad \text{(ER:Rule } 3^+\text{)}$$

Note that, as with Rule 1, Rule 3^+ is only a sufficient condition for P-domination.

Algorithm 2 shows how Phase I would look like when topological sort is applied and Rules 1 and 3^+ are used to filter out tuples. Tuples that survive after this phase are called *candidates*, with C^p denoting the set of all candidates.

Algorithm 2. PHASE I with Rules 1 and 3^+
Input: probabilistic relation R^p
Output: bounds $H_G^-(u)$ and $H_G^+(u)$ for each candidate tuple ($u \in C^p$)

```
1: topologically sort R^p                                   ▷ Phase I.a
2: for all tuples u ∈ R^p do                                ▷ Phase I.b
3:     H_G^-(u) ← 0, H_G^+(u) ← P − (P_{G_u} + p(u))
4:     u.filtered ← false
5:     for all tuples v ∈ R^p preceding u do
6:         if v ≻ u then
7:             if v ≻_p u [Rule 1] then u.filtered ← true
8:             if v.filtered = false ∧ G_u ≠ G_v then
9:                 H_G^+(v) ← H_G^+(v) − p(u)
10:            if u.filtered = false ∧ G_u ≠ G_v then
11:                H_G^-(u) ← H_G^-(u) + p(v)
12:        else if v ≻_p u [Rule 3^+] then u.filtered ← true
13:        if u ≻_p v [Rule 3^+] then v.filtered ← true
```

5.2 Phase II

In Phase II all P-domination rules that have not been applied in Phase I are used to filter out those candidate tuples in C^p that are not part of $\mathrm{Sky}(R^p)$. Two main issues are to be addressed in this phase: 1) Whether the rules are applied at the same time or not; and 2) how to efficiently evaluate Rule 2, which is the most costly one. As to the first point, we consider the possibility of a 2-scan approach: In the first scan of candidate tuples, only Rules 1 and 3 are applied (Rule 1 only if not already checked in Phase I), whereas the second scan applies Rule 2 to unfiltered tuples. The rationale for this approach is that, given the bounds computed in Phase I, Rule 3 has $\mathcal{O}(1)$ complexity, whereas a single evaluation of Rule 2 still requires $\mathcal{O}(N)$ time; thus, it is worthwhile to delay checking Rule 2 as much as possible. When tuples are topologically sorted, this strategy guarantees that Rule 2 is applied the minimum number of times.

Lemma 1. *If R^p is topologically sorted and Rule 2 is applied only after both Rules 1 and 3 have been tested on all pairs of tuples, then it will never compare two tuples none of which is part of the skyline of R^p.*

Proof. By contradiction, assume that Rule 2 is tested on tuples u and v, none of which belongs to $\mathrm{Sky}(R^p)$. Without loss of generality, assume that $u \succ v$. By hypothesis, there exists a skyline tuple t that P-dominates u and this P-domination is due to Rule 2, thus it is $t \succ u$. Since R^p is topologically sorted, t precedes u, thus u will be dropped before v is read, proving the assert. □

The evaluation of Rule 2 for tuples u and v, with $u \succ v$, needs quantities that depend on both u and v, thus they cannot be precomputed in Phase I. We show here how Rule 2 can be rewritten so as to guarantee its efficient computation. To this end, the quantities $N(u, v)$, $D(u, v)$, and $\Delta(u, v)$ are rewritten so as to avoid searching tuples that are indifferent to a given tuple; the rationale for this is to favor algorithms using either a spatial access method indexing the tuples in R^p or a topological sort of tuples in R^p. In the former case, retrieving all tuples dominated by a tuple u or that dominate u requires issuing a single window query, while accessing all tuples that are indifferent to u amounts to the resolution of 2^{d-1} window queries, where d is the number of skyline attributes; in the latter case, tuples that dominates u are guaranteed to precede u in the input relation. The straightforward (yet lengthy) algebraic transformations leading to the following Lemma are omitted for lack of space.

Lemma 2. *The quantities $D(u, v)$, $D(u, v) - \Delta(u, v)$, and $N(u, v) - \Delta(u, v)$ that appear in Rule 2 can be equivalently rewritten as, respectively, (for simplicity, we detail only the case when $G_u \neq G_v$):*

$$D(u, v) = P_{u,v} + 1 - P_{G_u} - H_G^-(v) + \sum_{u \succ t \succ v} p(t) + \left(\sum_{\substack{t \succ v \\ t \in G_u}} p(t) - \sum_{\substack{t \succ v \\ t \in G_v}} p(t) \right) - \sum_{\substack{u \succ t \succ v \\ t \in G_u}} p(t) - \sum_{\substack{t \sim u, t \sim v \\ t \in G_v}} p(t)$$

$$D(u,v) - \Delta(u,v) = P_{u,v} + 1 + P_{G_v} - 2P_{G_u} - (p(u) - p(v)) - H_G^+(u) - \left(\sum_{\substack{u \succ t \succ v \\ t \in G_u}} p(t) - \sum_{\substack{u \succ t \succ v \\ t \in G_v}} p(t) \right) + \left(\sum_{\substack{t \succ v \\ t \in G_u}} p(t) - \sum_{\substack{t \succ v \\ t \in G_v}} p(t) \right) + \left(\sum_{\substack{u \succ t \\ t \in G_u}} p(t) - \sum_{\substack{u \succ t \\ t \in G_v}} p(t) \right) + \left(\sum_{\substack{t \sim u, t \sim v \\ t \in G_u}} p(t) - \sum_{\substack{t \sim u, t \sim v \\ t \in G_v}} p(t) \right)$$

$$N(u,v) - \Delta(u,v) = P_{u,v} + 1 - (p(u) - p(v)) - P_{G_u} - H_G^+(u) - \sum_{u \succ t \succ v} p(t) + \sum_{\substack{u \succ t \\ t \in G_u}} p(t) - \sum_{\substack{u \succ t \\ t \in G_v}} p(t) + \sum_{\substack{u \succ t \succ v \\ t \in G_v}} p(t) + \sum_{\substack{t \sim u, t \sim v \\ t \in G_u}} p(t).$$

Terms appearing in above equations fall into one of the three following categories:

1. Terms that also appear in other rules and, as such, are already available, namely: $P_{u,v}$, P_{G_u}, P_{G_v}, and the bounds $H_G^-(v)$ and $H_G^+(u)$.
2. Sums that extend over tuples of a *single* group (either G_u or G_v); these can be efficiently evaluated thanks to the `groupsHT` hash table, that allows direct retrieval of all tuples in a group.
3. The sum of probabilities of all tuples t that are dominated by u and dominate v, i.e., $\texttt{WndPr}(u, v) \stackrel{\text{def}}{=} \sum_{u \succ t \succ v} p(t)$.

For computing the "window sum of probabilities" $\texttt{WndPr}(u, v)$ we consider three methods:

1. With a spatial access method, one can quickly retrieve all tuples in a box whose extreme vertices coincide with attribute values of v and u. In particular, we consider R-trees [29].[7]
2. If no index is used, yet R^p has been sorted in Phase I.a, all tuples contributing to $\texttt{WndPr}(u, v)$ will follow u and precede v in the order.
3. Finally, if R^p is unordered, we necessarily have to iterate through all tuples in R^p.

Algorithm 3 shows an instantiation of Phase II when two scans of candidates are performed, such tuples are unordered, Rule 1 has been already applied in Phase I, and an R-tree is built over R^p.

Algorithm 3. PHASE II with R-tree and 2 scans
Input: probabilistic relation R^p, candidate set $C^p \subseteq R^p$
Output: $\text{Sky}(R^p)$

```
 1: build an R-tree indexTree over R^p                ▷ Phase II.a
 2: Sky(R^p) ← ∅                                      ▷ Phase II.b
 3: for all tuples u ∈ C^p do                         ▷ 1st scan
 4:     for all tuples v ∈ C^p preceding u do
 5:         if u ≻_p v [Rule 3] then C^p ← C^p \ {v}
 6:         else if v ≻_p u [Rule 3] then C^p ← C^p \ {u}
 7: for all tuples u ∈ C^p do                         ▷ 2nd scan
 8:     insert ← true
 9:     for all tuples v ∈ Sky(R^p) do
10:         if u ≻ v then
11:             WndPr(u, v) ← SumPr(indexTree.
                WndQuery(u, v))
12:             if u ≻_p v [Rule 2] then Sky(R^p) ←
                Sky(R^p) \ {v}
13:         else if v ≻ u then
14:             WndPr(v, u) ← SumPr(indexTree.
                WndQuery(v, u))
15:             if v ≻_p u [Rule 2] then insert ← false, break
16:     if insert then Sky(R^p) ← Sky(R^p) ∪ {u}
```

7. We also used aR-trees [30], which are able to solve aggregate queries, but results (not shown here) were not encouraging.

TABLE 4
Settings for the Generation of Synthetic Data Sets

symbol	description	range	default
d	dimensionality	$[2-5]$	4
N	cardinality	$[10K-200K]$	100K
N_G	tuples per group	$[2-6]$	2
f_G	% of grouped tuples	$[10\%-50\%]$	10%

6 Experimental Evaluation

In this section, we experimentally analyze the efficiency of our algorithms, which we implemented in Java[8] and run on a Pentium IV 3.4-GHz PC equipped with 512 MB of main memory. For all experiments, we are mainly interested in overall clock time needed to compute the skyline. Supplemental statistics include the number of candidate tuples and the effectiveness of P-domination rules. Due to the modular nature of our algorithms, each of them can be identified as a 4-part name using the symbols in Table 3. For instance, the algorithm srt|R1|noidx|2scans will perform a topological sort of tuples (srt), apply only Rule 1 in Phase I (R1), will not build any spatial index (noidx) and will perform two scans over candidates in Phase II (2scans).

6.1 Experiments on Synthetic Data Sets

To assess the scalability and robustness of the algorithms with respect to different data characteristics, we used synthetic data sets, that allow to freely modify the different parameters used for data generation. In particular, we adapted to our scenario the procedure used in [18]. Once all the tuples have been generated according to a particular distribution, groups are formed as follows: A fraction f_G of tuples is involved in groups containing more than one tuple, while all other tuples will belong to "singleton" groups; each nonsingleton group is formed by exactly N_G tuples, randomly picked from the data set (this is repeated in case the probability of the group exceeds 1). We therefore have $f_G N / N_G$ nonsingleton groups and $(1 - f_G)N$ singletons. Table 4 shows the range of parameters used in the experiments; unless otherwise stated, default values and a uniform distribution for both data and probability are used.

8. The authors thank Marios Hadjieleftheriou for his R-tree code.

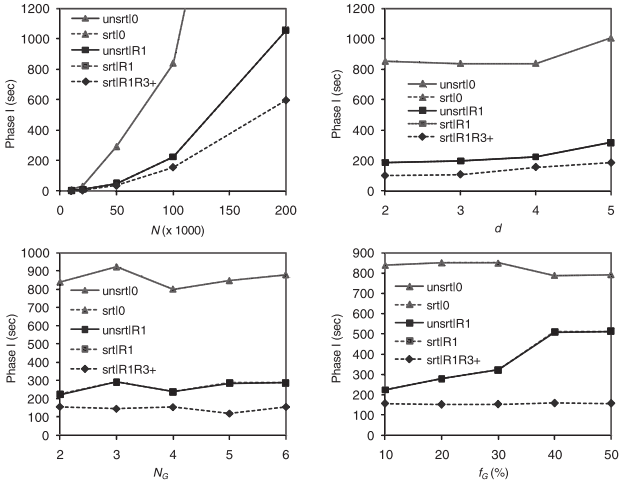

Fig. 5. Runtime for alternative Phase I strategies.

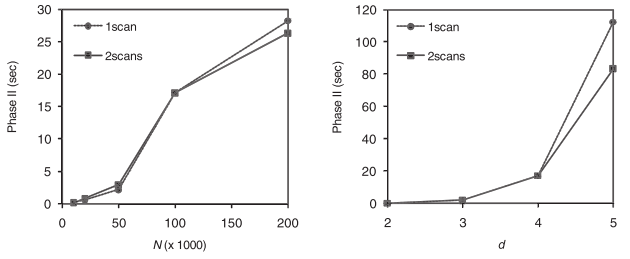

Fig. 6. Runtime for alternative Phase II.b strategies.

We first compare alternatives for Phase I. The choices are whether to sort data or not and which rules to use to filter out tuples. Fig. 5 shows running times for the five alternatives (unsrt | R1R3+is not a viable choice) when varying parameters in Table 4. It can be observed that sorting tuples does not improve the performance of $\emptyset$ and R1 alternatives (graphs of the srt and unsrt versions are almost coincident), whereas it is determinant in making srt|R1R3+ the best alternative in all considered setups. When no rule is used, the quadratic dependence on N is evident, whereas applying rules largely mitigates this effect. Finally, we observe that srt|R1R3+ is particularly robust to modifications of the N_G and f_G parameters.

Besides being the most efficient in computing bounds, the srt|R1R3+ variant is also able to filter out a high number of nonskyline tuples. Results (not shown here for brevity) demonstrate that this strategy is able to prune 98 percent of P-dominated tuples during Phase I, reaching 99.9 percent for the simplest $d = 2$ data set. This clearly also helps improving performance of Phase II, because only a limited number of candidate tuples are to be compared there. Considering all the above, the next experiments only consider srt|R1R3+ for implementing Phase I.

In Fig. 6, we compare alternatives for Phase II.b, when no index is used. We show only results when varying N and d, because no visible effect is observed when changing other parameters. It is worth noting that both 1scan and 2scans perform very well (running times are 1 order of magnitude lower than the time spent in Phase I). For high N and d values, the 2scans approach pays off, saving about 20 percent of running time when $d = 5$. In the following, therefore, we concentrate on the 2scans variant.

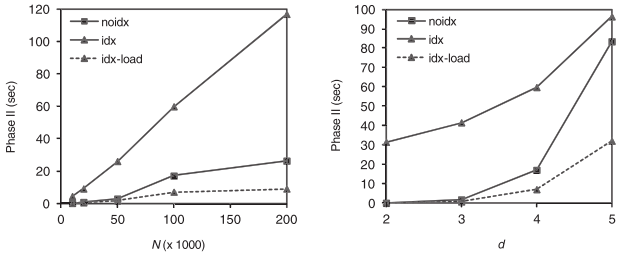

Fig. 7. Runtime for alternative Phase II.a strategies.

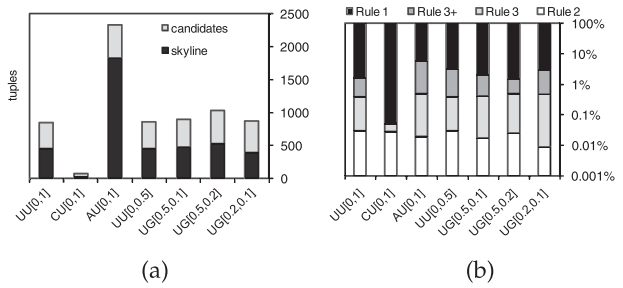

Fig. 8. (a) Skyline and candidates cardinality and (b) P-domination rules effectiveness.

In Fig. 7, we analyze the alternatives for Phase II.a of the algorithm. Since we already opted for a sort in Phase I, the possible choices are whether or not to use an index. The noidx alternative is clearly always the best solution, outperforming R-trees in all settings. However, notice that, if the cost for building the tree has not to be paid (idx-load curve in the figure), the running time of idx reduces to about 50 percent that of the noidx approach. Since both strategies are unaffected when varying either N_G or f_G, we do not show such results here.

In our last series of experiments, we vary the distribution of tuples and probabilities. In Fig. 8a, we show the cardinality of $\mathrm{Sky}(R^p)$ for different data/probability distributions. The two letters in each data set name correspond to data distribution (uniform, U, correlated, C, or anticorrelated, A) and probability distribution (uniform, U, or Gaussian, G); numbers in brackets report the probability range for U or mean and variance for G. The figure also shows the number of candidate tuples obtained when exploiting the srt|R1R3+ strategy during Phase I. For all considered distributions, the skyline is of manageable size, reaching a maximum of 1,832 tuples for the anticorrelated data set; moreover, the limited number of candidate tuples again demonstrates the effectiveness of srt|R1R3+. Fig. 8b reports in a logarithmic scale the pruning capabilities of P-domination rules, as applied by the srt|R1R3+|noidx|2scans algorithm. It is clear that the cheap Rule 1 is the most effective in filtering out tuples, whereas the costly Rule 2, which is the last one to be applied, only prunes a very low percentage of P-dominated tuples ($0.01\% \div 0.02\%$).

Finally, Table 5 compares running times of the baseline unsrt|$\emptyset$|noidx|1scan algorithm with the most efficient srt|R1R3+|noidx|2scans: For all data distributions, the latter attains costs which are 1 order of magnitude lower than the naïve strategy, the most difficult data set being the anticorrelated one.

TABLE 5
Runtime (secs) for the srt|R1R3+|noidx|2scans (and the unsrt|∅|noidx|1scan) Algorithms

Dataset	Phase I		Phase II		total	
UU[0,1]	156.2	(838.1)	17.1	(2106.0)	173.3	(2944.1)
CU[0,1]	130.4	(836.4)	0.3	(1470.7)	130.7	(2307.1)
AU[0,1]	202.3	(937.3)	137.1	(3522.3)	339.4	(4459.6)
UU[0,0.5]	227.8	(831.3)	20.9	(1801.9)	248.7	(2633.2)
UG[0.5,0.1]	176.6	(854.2)	27.0	(1895.5)	203.6	(2749.7)
UG[0.5,0.2]	181.7	(821.8)	16.4	(2524.8)	198.1	(3346.6)
UG[0.2,0.1]	124.4	(863.2)	20.9	(1562.7)	145.3	(2425.9)

TABLE 6
Runtime (msecs) and Skyline Size for Exp. Score

Dataset	time	$\lvert\mathrm{Sky}(R^p)\rvert$	Dataset	time	$\lvert\mathrm{Sky}(R^p)\rvert$
UU[0,1]	104	860	UG[0.5,0.1]	106	907
CU[0,1]	35	75	UG[0.5,0.2]	129	1033
AU[0,1]	641	2347	UG[0.2,0.1]	108	889
UU[0,0.5]	100	860			

We conclude our analysis on synthetic data sets by showing results for the expected score semantics. As already stated in Section 5, in this case the probabilistic skyline can be efficiently computed by appropriately modifying a "traditional" skyline algorithm; we show in Table 6 results obtained when applying the SFS algorithm [25] with sorting based on the sum of attributes and probability values.

6.2 Experiments on the NBA Data Set

Our next series of experiments concerns the real NBA data set: This data set contains 339,721 performances of 1,313 players during the seasons between 1991 and 2005. We only considered points, rebounds and assists, because other statistics (e.g., steals) are not available for each season. Each (unique) performance by a player is a tuple, while players represent groups: Each player (group) has a total probability of 1, thus if he played in m games each of his performances has a probability of $1/m$ (if a same performance has been repeated k times, the probability of such tuple is k/m).

Notice that this data set, which conforms to the x-relation model, was previously used by working at the *group level*, i.e., for determining the "best" players [19], [22]. We decided to use it also in our context, which works at the *tuple level* and is, therefore, able to derive which are the "best" performances, for several reasons. First, due to its size, it represents, regardless of the "level" at which one operates, a good benchmark for testing efficiency of skyline algorithms. Second, best performances can provide new insights on the data set. For instance, one could rank players by looking at how many of their performances appear in the skyline, multiple (possibly quite different) performances of a same player being in the skyline can be seen as an evidence that such player exhibits strong abilities in different aspects of the game, and so on. Finally, as discussed below, one can analyze how the skyline varies in the "transition" from the probabilistic to the deterministic (average) case.

Table 7 reports the performance of our best algorithm on the NBA data set (refer to the row $\varepsilon = 0$). The high running time for the expected rank semantics is essentially due to Phase I, which is unable to prune a high number of tuples. The reason of this behavior is that all groups have probability 1; thus, the rightmost part of Rule 1 becomes $p(u) \geq 1/2$, which can be true only for players with very few performances. Rule 1, therefore, holds very rarely (such players hardly have very good performances), and candidate tuples amount to about 50 percent of the data set. This is also to say that Phase I pays a quadratic cost. On the other hand, the skyline only contains 14 tuples and most candidate tuples are pruned by the cheap Rule 3. The cost for Phase II is, thus, almost negligible (~ 0.01 percent) with respect to that of Phase I.

TABLE 7
NBA: Skyline Size and Runtime (msecs) versus ε

ε	N	$\lvert\mathrm{Sky}(R^p)\rvert$ exp. rank		$\lvert\mathrm{Sky}(R^p)\rvert$ exp. score	
0	231225	14	(1525947)	323	(1609)
0.1	18924	91	(30604)	668	(335)
0.5	1516	29	(134)	110	(17)
0.9	1313	20	(46)	20	(8)

It is known that the ranking of tuples under the expected rank semantics is highly influenced by probability values [13]. This is confirmed by our results, because all the 14 tuples in the skyline have a (very) high probability, representing good performances of players with only a few played games. All the 65 performances that are in the skyline when regarding NBA as a deterministic relation are also present in the probabilistic case when using the expected score semantics. Expected score considers as optimal other 258 performances, thus leading to a total of 323 skyline tuples (23 times the expected rank skyline size). Many of these 258 performances, however, correspond to singleton groups, i.e., players with only one game, that are hardly prunable by the only P-domination rule of expected score semantics.

To try to gain some more insight on the relationship between the skyline of the NBA data set (consisting of single performances) with the one obtained from the "aggregate" case in which each player has a single tuple representing his average performance, we explore the effect of partially aggregating similar performances. To this end, performances are first normalized into the $[0, 1]$ range. Then, for each player, we randomly choose a tuple: Tuples of that player whose euclidean distance to the chosen tuple is not higher than a cluster radius ε are combined into a single "average" tuple for that player; the procedure is then repeated on remaining tuples, until all tuples have been included into a cluster. Thus, by increasing ε we are able to obtain a sequence of data sets of decreasing size, whose tuples represent clusters of a larger number of performances. At $\varepsilon = 0.9$, it is observed that all players are each represented by a single cluster, i.e., complete aggregation is reached.

Data set and skyline sizes obtained for different ε values are shown in Table 7. For the expected rank semantics, having $\varepsilon > 0$ drops from the skyline tuples of players with only a few played games, which were there at $\varepsilon = 0$ only because of their high probability and are now P-dominated by clusters of better players (if the probability of the cluster is sufficiently high). With respect to the fully aggregate

"average" case, intermediate ε values add to the skyline clusters of consistently good players, whereas players with only a limited number of good performances are dropped. For instance, at $\varepsilon = 0.5$ the skyline contains clusters for all the 20 best players in the average case but Tim Hardaway, who is represented by two clusters: The one of his best performances has only 8 percent probability, while the other is not as good and turns to be P-dominated by a cluster of Stephon Marbury. Clearly, performances of Tim Hardaway are not consistently outstanding; thus, they are not included in the skyline. On the other hand, the skyline at $\varepsilon = 0.5$ also contains clusters of 10 players, such as Mike Bibby, Scottie Pippen, and Dwyane Wade, who do not appear in the average case, yet they have consistently outstanding performances.

7 Conclusions

In this paper, we have introduced an original definition of domination among tuples of a probabilistic relation (*P-domination*), and proved that the skyline resulting from it preserves all the basic properties that hold in the deterministic case. In particular, the skyline is still equal to the union of all top-1 results when one considers all possible scoring functions that are monotone in the skyline attributes. Our definitions apply to any semantics for ranking probabilistic tuples and model of tuple correlation, and as such have general validity. We have also shown how P-domination can be efficiently checked by means of a set of rules, which for space reasons we have detailed only for the expected rank and the expected score ranking semantics. For efficiently computing the skyline we have then provided a family of algorithms based on a common 2-phase structure, and shown through experiments on large probabilistic data sets their improvement with respect to a naïve evaluation.

An interesting issue for future work is to devise P-domination rules and algorithms for correlation models more complex than the one considered in this paper, in which only mutual exclusion rules are possible. Further, studying how the skyline depends on the specific ranking semantics is a relevant issue from a practical point of view. Related to this is the problem of precisely characterizing the complexity of skyline queries with respect to that of the underlying ranking semantics and correlation model.

References

[1] M.J. Cafarella, C. Re, D. Suciu, and O. Etzioni, "Structured Querying of Web Text Data: A Technical Challenge," *Proc. Conf. Innovative Data Systems Research (CIDR)*, 2007.

[2] X.L. Dong, A.Y. Halevy, and C. Yu, "Data Integration with Uncertainty," *Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB)*, 2007.

[3] R. Cheng, D.V. Kalashnikov, and S. Prabhakar, "Querying Imprecise Data in Moving Object Environments," *IEEE Trans. Knowledge Data Eng.*, vol. 16, no. 9, pp. 1112-1127, Sept. 2004.

[4] R. Cheng, D.V. Kalashnikov, and S. Prabhakar, "Evaluating Probabilistic Queries over Imprecise Data," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2003.

[5] P. Agrawal, O. Benjelloun, A.D. Sarma, C. Hayworth, S.U. Nabar, T. Sugihara, and J. Widom, "Trio: A System for Data, Uncertainty, and Lineage," *Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB)*, 2006.

[6] A.D. Sarma, O. Benjelloun, A.Y. Halevy, and J. Widom, "Working Models for Uncertain Data," *Proc. IEEE 22nd Int'l Conf. Data Eng. (ICDE)*, 2006.

[7] L. Antova, T. Jansen, C. Koch, and D. Olteanu, "Fast and Simple Relational Processing of Uncertain Data," *Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE)*, 2008.

[8] N.N. Dalvi and D. Suciu, "Efficient Query Evaluation on Probabilistic Databases," *Proc. Int'l Conf. Very Large Data Bases (VLDB)*, 2004.

[9] S. Börzsönyi, D. Kossmann, and K. Stocker, "The Skyline Operator," *Proc. IEEE 17th Int'l Conf. Data Eng. (ICDE)*, 2001.

[10] H. Chen, S. Zhou, and J. Guan, "Towards Energy-Efficient Skyline Monitoring in Wireless Sensor Networks," *Proc. Fourth European Conf. Wireless Sensor Networks (EWSN)*, 2007.

[11] I. Bartolini, Z. Zhang, and D. Papadias, "Collaborative Filtering with Personalized Skylines," *IEEE Trans. Knowledge Data Eng.*, vol. 23, no. 2, pp. 190-203, Feb. 2011.

[12] M.A. Soliman, I.F. Ilyas, and K.C.-C. Chang, "Top-k Query Processing in Uncertain Databases," *Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE)*, 2007.

[13] J. Li, B. Saha, and A. Deshpande, "A Unified Approach to Ranking in Probabilistic Databases," *Proc. Int'l Conf. Very Large Data Bases (VLDB)*, 2009.

[14] X. Zhang and J. Chomicki, "On the Semantics and Evaluation of Top-k Queries in Probabilistic Databases," *Proc. Int'l Workshop Ranking in Databases (DBRank)*, 2008.

[15] G. Cormode, F. Li, and K. Yi, "Semantics of Ranking Queries for Probabilistic Data and Expected Ranks," *Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE)*, 2009.

[16] I. Bartolini, P. Ciaccia, and M. Patella, "The Skyline of a Probabilistic Relation," DEIS, TR DEIS-MMDBG-11-10, http://www-db.deis.unibo.it/research/papers/TRs/DEIS-MMDBG-11-10.pdf, 2011.

[17] R. Fagin, A. Lotem, and M. Naor, "Optimal Aggregation Algorithms for Middleware," *Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS)*, 2001.

[18] K. Yi, F. Li, G. Kollios, and D. Srivastava, "Efficient Processing of Top-K Queries in Uncertain Databases with X-Relations," *IEEE Trans. Knowledge Data Eng.*, vol. 20, no. 12, pp. 1669-1682, Dec. 2008.

[19] J. Pei, B. Jiang, X. Li, and Y. Yuan, "Probabilistic Skylines on Uncertain Data," *Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB)*, 2007.

[20] M.J. Atallah and Y. Qi, "Computing all Skyline Probabilities for Uncertain Data," *Proc. 28th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS)*, 2009.

[21] W. Zhang, X. Lin, Y. Zhang, W. Wang, and J.X. Yu, "Probabilistic Skyline Operator over Sliding Windows," *Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE)*, 2009.

[22] X. Lin, Y. Zhang, W. Zhang, and M.A. Cheema, "Stochastic skyline Operator," *Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE)*, 2011.

[23] E. Szpilrajn, "Sur l'extension de l'ordre Partiel," *Fundamenta Mathematicae*, vol. 16, pp. 386-389, 1930.

[24] D. Yan and W. Ng, "Robust Ranking of Uncertain Data," *Proc. 16th Int'l Conf. Database Systems for Advanced Applications (DASFAA)*, 2011.

[25] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with Presorting," *Proc. IEEE 19th Int'l Conf. Data Eng. (ICDE)*, 2003.

[26] I. Bartolini, P. Ciaccia, and M. Patella, "Efficient Sort-Based Skyline Evaluation," *ACM Trans. Database Systems*, vol. 33, no. 4, article 31, 2008.

[27] H.T. Kung, F. Luccio, and F.P. Preparata, "On Finding the Maxima of a Set of Vectors," *J. ACM*, vol. 22, no. 4, pp. 469-476, 1975.

[28] C. Sheng and Y. Tao, "On Finding Skylines in External Memory," *Proc. 30th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS)*, 2011.

[29] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 1984.

[30] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP Operations in Spatial Data Warehouses," *Proc. Seventh Int'l Symp. Advances in Spatial and Temporal Databases (SSTD)*, 2001.

Ilaria Bartolini received the graduate degree in computer science and the PhD degree in electronic and computer engineering from the University of Bologna. She is currently an assistant professor at the University of Bologna, Italy, with DEIS. She has been a visiting researcher at CWI, Amsterdam, NJIT, Newark, New Jersey, and HKUST, Hong Kong. Her current research interests include collaborative filtering, similarity and preference-based query processing, and retrieval in multimedia collections. She is a member of the IEEE.

Paolo Ciaccia received the "Laurea" degree in electronic engineering and the PhD degree in electronic and computer engineering from the University of Bologna, where he has been a full professor since 2000. His current research interests include similarity and preference-based query processing, multimedia systems, and probabilistic databases. He was an associate editor of *IEEE Transactions on Knowledge and Data Engineering* and currently serves on the editorial board of the *ACM Transactions on Database Systems*. He is a member of the IEEE.

Marco Patella received the "Laurea" degree in electronic engineering and the PhD degree in electronic and computer engineering from the University of Bologna, Italy. Since 2006, he has been an associate professor at the University of Bologna with DEIS. His current research interests include similarity- and preference-based query processing in multimedia databases. He is one of the designers of the M-tree, an index for metric data.