On the Semantics and Evaluation of Top-k Queries in Probabilistic Databases

Xi Zhang
University at Buffalo, SUNY
xizhang@cse.buffalo.edu

Jan Chomicki
University at Buffalo, SUNY
chomicki@cse.buffalo.edu

Abstract

We formulate three intuitive semantic properties for top-k queries in probabilistic databases, and propose Global-Topk query semantics which satisfies all of them. We provide a dynamic programming algorithm to evaluate top-k queries under Global-Topk semantics in simple probabilistic relations. For general probabilistic relations, we show a polynomial reduction to the simple case. Our analysis shows that the complexity of query evaluation is linear in k and at most quadratic in database size.

1. Introduction

The study of incompleteness and uncertainty in databases has long been an interest of the database community [14, 5, 11, 1, 9, 23, 15]. Recently, this interest has been rekindled by an increasing demand for managing rich data, often incomplete and uncertain, emerging from scientific data management, sensor data management, data cleaning, information extraction etc. [6] focuses on query evaluation in traditional probabilistic databases; ULDB [3] supports uncertain data and data lineage in Trio [21]; MayBMS [17] uses the vertical World-Set representation of uncertain data [2]. The standard semantics adopted in most works is the *possible worlds* semantics [14, 9, 23, 3, 6, 2].

On the other hand, since the seminal papers of Fagin [7, 8], the top-k problem has been extensively studied in multimedia databases [18], middleware systems [16], data cleaning [10], core technology in relational databases [12, 13] etc. In the top-k problem, each tuple is given a *score*, and users are interested in k tuples with the highest scores.

More recently, the top-k problem has been studied in probabilistic databases [20, 19]. Those papers, however, are solving two essentially different top-k problems. Soliman et al. [20] assumes the existence of a scoring function to rank tuples. Probabilities provide information on how likely tuples will appear in the database. In contrast, in [19], the ranking criterion for top-k is the probability associated with each query answer. In many applications, it is necessary to deal with tuple probabilities and scores at the same time. Thus, in this paper, we use the model of [20]. Even in this model, different semantics for top-k queries are possible, so a part of the challenge is to define a reasonable semantics.

As a motivating example, considering the smart environment scenario in Example 1.

Example 1. *A smart lab has the following data for a Saturday night.*

Name	*Biometric Score (Face, Voice, ...)*	*Prob. of Sat Nights*
Aidan	0.65	0.3
Bob	0.55	0.9
Chris	0.45	0.4

Typically, the lab collects two kinds of data:

- *Biometric data from sensors;*
- *Historical statistics.*

Biometric data come from the sensors deployed in the lab, for example, face recognition and voice recognition sensors. Those data are collected and matched against the profile of each person involved in the lab. They can be normalized to give us an idea of how well each person fits the sensed data. In addition, the lab also keeps track of the statistics of each person's activities.

Knowing that we definitely had two visitors that night, we would like to ask the following question:

"Who were the two visitors in the lab last Saturday night?"

This question can be formulated as a top-k query over the above probabilistic relation, where $k = 2$.

In Example 1, each tuple is associated with an *event*, which is that candidate being in the lab on Saturday nights. The probability of the event is shown next to each tuple. In this example, all the events of tuples are independent, and tuples are therefore said to be *independent*. Intuitively, for the top-k problem in Example 1, we are not necessarily interested in candidates with high biometric scores if the associated events are very unlikely to happen, e.g. we have strong evidence suggesting that a candidate plays football on Saturday nights and his probability of being in lab is 0.001.

Example 1 shows a *simple* probabilistic relation where the tuples are independent. In contrast, Example 2 illustrates a more general case.

Example 2. *In a sensor network deployed in a habitat, each sensor reading comes with a confidence value* Prob*, which is the probability that the reading is valid. The following table shows the temperature sensor readings at a given sampling time. These data are from two sensors, Sensor 1 and Sensor 2, which correspond to two* parts *of the relation, marked C_1 and C_2 respectively. Each sensor has only one* true *reading at a given time, therefore tuples from the same part of the relation correspond to exclusive events.*

	Temp.(°F)	*Prob*
C_1	22	0.6
	10	0.4
C_2	25	0.1
	15	0.6

Our question is:

"What's the temperature of the warmest spot?"

The question can be formulated as a top-k query, where $k = 1$, over a probabilistic relation containing the above data. However, we must take into consideration that the tuples in each part $C_i, i = 1, 2$, are exclusive.

Our contributions in this paper are the following:

- We formulate three intuitive semantic properties and use them to compare different top-k semantics in probabilistic databases (Section 3.1);
- We propose a new semantics for top-k queries in probabilistic databases, called Global-Top$k$, which satisfies all the properties above (Section 3.2);
- We exhibit efficient algorithms for evaluating top-k queries under the Global-Top$k$ semantics in *simple* probabilistic databases (Section 4.1) and general probabilistic databases (Section 4.3).

2. Background

Probabilistic Relations To simplify the discussion in this paper, a probabilistic database contains a single *probabilistic relation*. We refer to a traditional database relation as a *deterministic relation*. A deterministic relation R is a set of tuples. A *partition* $\mathcal{C}$ of R is a collection of non-empty subsets of R such that every tuple belongs to one and only one of the subsets. That is, $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ such that $C_1 \cup C_2 \cup \ldots \cup C_m = R$ and $C_i \cap C_j = \emptyset, 1 \leq i \neq j \leq m$. Each subset $C_i, i = 1, 2, \ldots, m$ is a *part* of the partition $\mathcal{C}$. A *probabilistic relation* R^p has three components, a *support (deterministic) relation* R, a probability function p and a partition $\mathcal{C}$ of the support relation R. The probability function p maps every tuple in R to a probability value in $(0, 1]$. The partition $\mathcal{C}$ divides R into subsets such that the tuples within each subset are exclusive and therefore their probabilities sum up to at most 1. In the graphical presentation of R, we use horizontal lines to separate tuples from different parts.

Definition 2.1 (Probabilistic Relation)**.** *A probabilistic relation R^p is a triplet $\langle R, p, \mathcal{C}\rangle$, where R is a support deterministic relation, p is a probability function $p : R \mapsto (0, 1]$ and $\mathcal{C}$ is a partition of R such that $\forall C_i \in \mathcal{C}, \sum_{t \in C_i} p(t) \leq 1$.*

In addition, we make the assumption that tuples from different parts of of $\mathcal{C}$ are independent, and tuples within the same part are exclusive. Def. 2.1 is equivalent to the model used in Soliman et al. [20] with exclusive tuple generation rules. Ré et al. [19] proposes a more general model, however only a restricted model equivalent to Def. 2.1 is used in top-k query evaluation.

Example 2 shows an example of a probabilistic relation whose partition has two parts. Generally, each part corresponds to a real world entity, in this case, a sensor. Since there is only one true state of an entity, tuples from the same part are exclusive. Moreover, the probabilities of all possible states of an entity sum up to at most 1. In Example 2, the sum of probabilities of tuples from Sensor 1 is 1, while that from Sensor 2 is 0.7. This can happen for various reasons. In the above example, we might encounter a physical difficulty in collecting the sensor data, and end up with partial data.

Definition 2.2 (Simple Probabilistic Relation)**.** *A probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$ is simple iff the partition $\mathcal{C}$ contains only singleton sets.*

The probablistic relation in Example 1 is simple (individual parts not illustrated). Note that in this case, $|R| = |\mathcal{C}|$.

We adopt the well-known *possible worlds* semantics for probabilistic relation [14, 9, 23, 3, 6, 2].

Definition 2.3 (Possible World)**.** *Given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$, a deterministic relation W is a* possible world *of R^p iff*

1. *W is a subset of the support relation, i.e. $W \subseteq R$;*
2. *For every part C_i in the partition $\mathcal{C}$, at most one tuple from C_i is in W, i.e. $\forall C_i \in \mathcal{C}, |C_i \cap W| \leq 1$.*

Denote by $pwd(R^p)$ *the set of all possible worlds* of R^p. Since all the parts in $\mathcal{C}$ are independent, we have the following definition of the probability of a possible world.

Definition 2.4 (Probability of a Possible World)**.** *Given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$, for any $W \in pwd(R^p)$, its probability $Pr(W)$ is defined as*

$$Pr(W) = \prod_{t \in W} p(t) \prod_{C_i \in \mathcal{C}'} (1 - \sum_{t \in C_i} p(t)) \qquad (1)$$

where $\mathcal{C}' = \{C_i \in \mathcal{C} | W \cap C_i = \emptyset\}$.

Scoring function A *scoring function over a deterministic relation* R is a function from R to real numbers, i.e. $s : R \mapsto \mathbb{R}$. The function s induces a *preference relation* $\succ_s$ and an *indifference relation* $\sim_s$ on R. For any two distinct tuples t_i and t_j from R,

$$t_i \succ_s t_j \text{ iff } s(t_i) > s(t_j);$$
$$t_i \sim_s t_j \text{ iff } s(t_i) = s(t_j).$$

When the scoring function s is *injective*, $\succ_s$ establishes a *total order*[1] over R. In such a case, no two tuples from R tie in score.

A *scoring function over a probabilistic relation* $R^p = \langle R, p, \mathcal{C}\rangle$ is a scoring function s over its support relation R.

Top-k Queries

Definition 2.5 (Top-k Answer Set over Deterministic Relation)**.** *Given a deterministic relation R, a non-negative integer k and a scoring function s over R, a top-k answer in R under s is a set T of tuples such that*

1. $T \subseteq R$;

2. If $|R| < k$, $T = R$, otherwise $|T| = k$;

3. $\forall t \in T, \forall t' \in R - T, t \succ_s t'$ or $t \sim_s t'$.

According to Def. 2.5, given k and s, there can be more than one top-k answer set in a deterministic relation R. The evaluation of a top-k query over R returns one of them nondeterministically, say S. However, if the scoring function s is injective, S is unique, denoted by $top_{k,s}(R)$.

3. Semantics of Top-k Queries

3.1. Semantic Properties of Top-k Answers

Probability opens the gates for various possible semantics for top-k queries. As the semantics of a probabilistic relation involves a set of worlds, it is to be expected that there may be more than one top-k answer, even under an injective scoring function. The answer to a top-k query over a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$ should clearly be a set of tuples from its support relation R. In order to compare different semantics, we formulate below some *properties* we would like any reasonable semantics to satisfy.

In the following discussion, S is any top-k answer set in $R^p = \langle R, p, \mathcal{C}\rangle$ under an injective scoring function s. A tuple from the support relation R is a *winner* if it belongs to some top-k answer set under that semantics, and a *loser* otherwise. That is to say, in the case of multiple top-k answer sets, any tuple from any of them is a winner.

Properties

1. *Exact k*: When R^p is sufficiently large ($|\mathcal{C}| \geq k$), the cardinality of S is exactly k;
2. *Faithfulness*: For any two tuples $t_1, t_2 \in R$, if both the score and the probability of t_1 are higher than those of t_2 and $t_2 \in S$, then $t_1 \in S$;
3. *Stability*:
   - Raising the score/probability of a winner will not turn it into a loser;
   - Lowering the score/probability of a loser will not turn it into a winner.

All of those properties reflect basic intuitions about top-k answers. *Exact k* expresses user expectations about the size of the result. *Faithfulness* and *Stability* reflect the significance of score and probability.

3.2. Global-Top$k$ Semantics

We propose here a new top-k answer semantics in probabilistic relations, namely *Global-Topk*, which satisfies all the properties formulated in Section 3.1:

- **Global-Top$k$**: return k highest-ranked tuples according to their probability of being in the top-k answers in possible worlds.

Considering a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$ under an injective scoring function s, any $W \in pwd(R^p)$ has a unique top-k answer set $top_{k,s}(W)$. Each tuple from the support relation R can be in the top-k answer (in the sense of Def. 2.5) in zero, one or more possible worlds of R^p. Therefore, the sum of the probabilities of those possible worlds provides a global ranking criterion.

Definition 3.1 (Global-Top$k$ Probability)**.** *Assume a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$, a non-negative integer k and an injective scoring function s over R^p. For any tuple t in R, the Global-Topk probability of t, denoted by $P^{R^p}_{k,s}(t)$, is the sum of the probabilities of all possible worlds of R^p whose top-k answer contains t.*

$$P^{R^p}_{k,s}(t) = \sum_{\substack{W \in pwd(R^p) \\ t \in top_{k,s}(W)}} Pr(W).$$

For simplicity, we skip the superscript in $P^{R^p}_{k,s}(t)$, i.e. $P_{k,s}(t)$, when the context is unambiguous.

Definition 3.2 (Global-Top$k$ Answer Set over Probabilistic Relation)**.** *Given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$, a non-negative integer k and an injective scoring function s over R^p, a top-k answer in R^p under s is a set T of tuples such that*

1. $T \subseteq R$;

2. If $|R| < k$, $T = R$, otherwise $|T| = k$;

3. $\forall t \in T, \forall t' \in R - T, P_{k,s}(t) \geq P_{k,s}(t')$.

Notice the similarity between Def. 3.2 and Def. 2.5. In fact, the probabilistic version only changes the last condition, which restates the preferred relationship between two tuples by taking probability into account. This semantics preserves the nondeterministic nature of Def. 2.5. For example, if two tuples are of the same Global-Top$k$ probability, and there are $k-1$ tuples with higher Global-Top$k$ probability, Def. 2.5 allows one of the two tuples to be added to

[1] irreflective, transitive, connected binary relation.

the top-k answer nondeterministically. Example 3 gives an example of the Global-Top$k$ semantics.

Example 3. *Consider the top-2 query in Example 1. Clearly, the scoring function here is the biometric scoring function. The following table shows all the possible worlds and their probabilities. For each world, the underlined people are in the top-2 answer set of that world.*

Possible World	*Prob*
$W_1 = \emptyset$	0.042
$W_2 = \{\underline{Aidan}\}$	0.018
$W_3 = \{\underline{Bob}\}$	0.378
$W_4 = \{\underline{Chris}\}$	0.028
$W_5 = \{\underline{Aidan}, \underline{Bob}\}$	0.162
$W_6 = \{\underline{Aidan}, \underline{Chris}\}$	0.012
$W_7 = \{\underline{Bob}, \underline{Chris}\}$	0.252
$W_8 = \{\underline{Aidan}, \underline{Bob}, Chris\}$	0.108

Chris is in the top-2 answer of W_4, W_6, W_7, so its top-2 probability is $0.028 + 0.012 + 0.252 = 0.292$. Similarly, the top-2 probability of Aidan and Bob are 0.9 and 0.3 respectively. $0.9 > 0.3 > 0.292$, therefore Global-Top$k$ will return $\{Aidan, Bob\}$.

Note that top-k answer sets may be of cardinality less than k for some possible worlds. We refer to such possible worlds as *small* worlds. In Example 3, $W_{1...4}$ are all small worlds.

3.3. Other Semantics

Soliman et al. [20] proposes two semantics for top-k queries in probabilistic relations.

- *U-Top$k$*: return the most probable top-k answer set that belongs to possible world(s);
- *U-k$Ranks$*: for $i = 1, 2, \ldots, k$, return the most probable i^{th}-ranked tuples across all possible worlds.

Example 4. *Continuing Example 3, under U-Top$k$ semantics, the probability of top-2 answer set $\{Bob\}$ is 0.378, and that of $\{Aidan, Bob\}$ is $0.162 + 0.108 = 0.27$. Therefore, $\{Bob\}$ is more probable than $\{Aidan, Bob\}$ under U-Top$k$. In fact, $\{Bob\}$ is the most probable top-2 answer set in this case, and will be returned by U-Top$k$.*

Under U-k$Ranks$ semantics, Aidan is in 1^{st} place in the top-2 answer of W_2, W_5, W_6, W_8, therefore the probability of Aidan being in 1^{st} place in the top-2 answers in possible worlds is $0.018 + 0.162 + 0.012 + 0.108 = 0.3$. However, Aidan is not in 2^{nd} place in the top-2 answer of any possible world, therefore the probability of Aidan being in 2^{nd} place is 0. In fact, we can construct the following table.

	Aidan	*Bob*	*Chris*
Rank 1	0.3	$\underline{0.63}$	0.028
Rank 2	*0*	$\underline{0.27}$	0.264

U-k$Ranks$ selects the tuple with the highest probability at each rank (underlined) and takes the union of them. In this example, Bob wins at both Rank 1 and Rank 2. Thus, the top-2 answer returned by U-k$Ranks$ is $\{Bob\}$.

The properties introduced in Section 3.1 lay the ground for comparing different semantics. In Table 1, a single "✓" (resp. "×") indicates that property is (resp. is not) satisfied under that semantics. "✓/×" indicates that, the property is satisfied by that semantics in *simple* probabilistic relations, but not in the general case.

Semantics	Exact k	Faithfulness	Stability
Global-Top$k$	✓	✓	✓
U-Top$k$	×	✓/×	✓
U-k$Ranks$	×	×	×

Table 1. Property satisfaction for different semantics

Global-Top$k$ satisfies all the properties while neither of the other two semantics does. For *Exact k*, Global-Top$k$ is the only one that satisfies this property. Example 4 illustrates the case when both U-Top$k$ and U-k$Ranks$ violate this property. It is not satisfied by U-Top$k$ because a *small* possible world with high probability could dominate other worlds. In that case, the dominating possible world might not have enough tuples. It is also violated by U-k$Ranks$ because a single tuple can win at multiple ranks in U-k$Ranks$. For *Faithfulness*, since U-Top$k$ requires all tuples in a top-k answer set to be compatible, this property can be violated when a high-score/probability tuple could be dragged down arbitrarily by its compatible tuples if they are not very likely to appear. U-k$Ranks$ violates both *Faithfulness* and *Stability*. Under U-k$Ranks$, instead of a set, a top-k answer is an ordered vector, where ranks are significant. A change in a tuple's probability/score might have unpredictable consequence on ranks, therefore those two properties are not guaranteed to hold.

4. Query Evaluation under Global-Top$k$

4.1. Simple Probabilistic Relations

We first consider a *simple* probabilistic relation $R^p = \langle R, p, \mathcal{C} \rangle$ under an injective scoring function s.

Theorem 4.1. *Given a simple probabilistic relation $R^p = \langle R, p, \mathcal{C} \rangle$ and an injective scoring function s over R^p, if $R = \{t_1, t_2, \ldots, t_n\}$ and $t_1 \succ_s t_2 \succ_s \ldots \succ_s t_n$, the following recursion on Global-Top$k$ queries holds.*

$$q(k,i) = \begin{cases} 0 & k = 0 \\ p(t_i) & 1 \leq i \leq k \\ (q(k, i-1)\frac{\bar{p}(t_{i-1})}{p(t_{i-1})} + q(k-1, i-1))p(t_i) & \\ & otherwise \end{cases} \quad (2)$$

where $q(k,i) = P_{k,s}(t_i)$ and $\bar{p}(t_{i-1}) = 1 - p(t_{i-1})$.

Proof. *See* Appendix.

Example 5. *Consider a simple probabilistic relation* $R^p = \langle R, p, \mathcal{C}\rangle$*, where* $R = \{t_1, t_2, t_3, t_4\}$*,* $p(t_i) = p_i, 1 \le i \le 4$*,* $\mathcal{C} = \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}\}$ *and an injective scoring function* s *such that* $t_1 \succ_s t_2 \succ_s t_3 \succ_s t_4$*. The following table shows the Global-Topk of* t_i*, where* $0 \le k \le 2$*.*

k	t_1	t_2	t_3	t_4
0	0	0	0	0
1	p_1	$\bar{p}_1 p_2$	$\bar{p}_1\bar{p}_2 p_3$	$\bar{p}_1\bar{p}_2\bar{p}_3 p_4$
2	$\mathbf{p_1}$	$\mathbf{p_2}$	$(\mathbf{\bar{p}_2 + \bar{p}_1 p_2})\mathbf{p_3}$	$((\mathbf{\bar{p}_2 + \bar{p}_1 p_2})\mathbf{\bar{p}_3} + \mathbf{\bar{p}_1\bar{p}_2 p_3})\mathbf{p_4}$

Row 2 (bold) is each t_i*'s Global-Top*2 *probability. Now, if we are interested in top-*2 *answer in* R^p*, we only need to pick the two tuples with the highest value in Row 2.*

Given the recursion in Thm. 4.1, we can apply the standard dynamic programming (DP) technique, together with a priority queue, to select k tuples with the highest Global-Top$k$ probability, as shown in Alg. 1. It is a one-pass computation on the probabilistic relation, which can be easily implemented even if secondary storage is used. The overhead is the initial sorting cost (not shown in Alg. 1), which would be amortized by the workload of consecutive top-k queries.

Algorithm 1 (Ind_Topk) Evaluate Global-Top$k$ queries in a Simple Probabilistic Relation

Require: $R^p = \langle R, p, \mathcal{C}\rangle, k$
1: Initialize a fixed cardinality $(k+1)$ priority queue Ans of $\langle t, prob\rangle$ pairs, which compares pairs on $prob$;
2: $q(0,1) = 0$;
3: **for** $k' = 1$ to k **do**
4: $\quad q(k', 1) = p(t_{k'})$;
5: **end for**
6: **for** $i = 2$ to $|R|$ **do**
7: $\quad$ **for** $k' = 0$ to k **do**
8: $\quad\quad$ **if** $k' = 0$ **then**
9: $\quad\quad\quad q(k', i) = 0$;
10: $\quad\quad$ **else**
11: $\quad\quad\quad q(k', i) = p(t_i)(q(k', i-1)\frac{\bar{p}(t_{i-1})}{p(t_{i-1})} + q(k'-1, i-1))$;
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: $\quad$ Add $\langle t_i, q(k,i)\rangle$ to Ans;
15: $\quad$ **if** $|Ans| > k$ **then**
16: $\quad\quad$ remove the pair with the smallest $prob$ value from Ans;
17: $\quad$ **end if**
18: **end for**
19: **return** $\{t_i | \langle t_i, q(k,i)\rangle \in Ans\}$;

4.2. Threshold Algorithm Optimization

Fagin [8] proposes *Threshold Algorithm(*TA*)* for processing top-k queries in a middleware scenario. In a middleware system, an *object* has m attributes. For each attribute, there is a sorted list ranking objects in the decreasing order of its score on that attribute. An *aggregation function* f combines the individual attribute scores x_i, i=$1, 2, \ldots, m$ to obtain the overall object score $f(x_1, x_2, \ldots, x_m)$. An aggregation function is *monotonic* iff $f(x_1, x_2, \ldots, x_m) \le f(x'_1, x'_2, \ldots, x'_m)$ whenever $x_i \le x'_i$ for every i. Fagin [8] shows that TA is cost-optimal in finding the top-k objects in such a system.

TA is guaranteed to work as long as the aggregation function is monotonic. For a simple probabilistic relation, if we regard *score* and *probability* as two special attributes, Global-Top$k$ probability $P_{k,s}$ is an aggregation function of *score* and *probability*. The *Faithfulness* property in Section 3.1 implies the monotonicity of Global-Top$k$ probability. Consequently, assuming that we have an index on probability as well, we can guide the dynamic programming(DP) in Alg. 1 by TA. Now, instead of computing all kn entries for DP, where $n = |R|$, the algorithm can be stopped as early as possible. A subtlety is that Global-Top$k$ probability $P_{k,s}$ is *only* well-defined for $t \in R$, unlike in [8], where an aggregation function is well-defined over the domain of all possible attribute values. Therefore, compared to the original TA, we need to achieve the same behavior without referring to virtual tuples which are not in R.

U-Top$k$ satisfies *Faithfulness* in simple probabilistic relations, but the candidates under U-Top$k$ are *sets* not *tuples* and thus there is no counterpart of an aggregation function under U-Top$k$. Therefore, TA is not applicable. Neither is TA applicable to U-kRanks. Though we can define an aggregation function per $rank$, $rank = 1, 2, \ldots, k$, for tuples under U-kRanks, the violation of *Faithfulness* in Table 1 suggests a violation of monotonicity of those k aggregation functions.

Denote T and P for the list of tuples in the decreasing order of score and probability respectively. Following the convention in [8], $\underline{t}$ and $\underline{p}$ are the last value seen in T and P respectively.

Applying TA to Global-Topk Computation.

(1) Go down T list, and fill in entries in the DP table. Specifically, for $\underline{t} = t_j$, compute the entries in the j^{th} column up to the k^{th} row. Add t_j to the top-k answer set Ans, if any of the following conditions holds:
   (a) Ans has less than k tuples, i.e. $|Ans| < k$;
   (b) The Global-Top$k$ probability of t_j, i.e. $q(k, j)$, is greater than the lower bound of Ans, i.e. LB_{Ans}, where $LB_{Ans} = \min_{t_i \in Ans} q(k, i)$.

   In the second case, we also need to drop the tuple with the lowest Global-Top$k$ probability in order to keep the cardinality of Ans.

(2) After we have seen at least k tuples in T, we go down P list to find the first p whose tuple t has not been seen.

Let $\underline{p} = p$, and we can use $\underline{p}$ to estimate the *threshold*, i.e. upper bound (UP) of the Global-Top$k$ probability of any unseen tuple. Assume $\underline{t} = t_i$,

$$UP = (q(k,i)\frac{\bar{p}(t_i)}{p(t_i)} + q(k-1,i))\underline{p}.$$

(3) If $UP > LB_{Ans}$, we can expect Ans will be updated in the future, so go back to (1). Otherwise, we can safely stop and report Ans.

Theorem 4.2 (Correctness)**.** *Given a simple probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$, a non-negative integer k and an injective scoring function s over R^p, the above TA-based algorithm correctly find a top-k answer under Global-Top$k$ semantics.*

Proof. *See* Appendix.

The optimization above aims at an early stop. Bruno et al. [4] carries out an extensive experimental study on the effectiveness of applying TA in RDMBS. They consider various aspects of query processing. One of their conclusions is that if at least one of the indices available for the attributes[2] is a *covering index*, that is, it is defined over all other attributes and we can get the values of all other attributes directly without performing a primary index lookup, then the improvement by TA can be up to two orders of magnitude. The cost of building a useful set of indices once would be amortized by a large number of top-k queries that subsequently benefit form such indices. Even in the lack of covering indices, if the data is highly correlated, in our case, that means high-score tuples having high probabilities, TA would still be effective.

4.3. Arbitrary Probabilistic Relations

Induced Event Relation In the general case of probabilistic relation, each part of of the partition $\mathcal{C}$ can contain more than one tuple. The crucial *independence* assumption in Alg. 1 no longer holds. However, even though tuples are not independent, *parts* of the partition $\mathcal{C}$ are. In the following definition , we assume an identifier function id. For any tuple t, $id(t)$ is the identifier of the part where t belongs.

Definition 4.1 (Induced Event Relation)**.** *Given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$, an injective scoring function s over R^p and a tuple $t \in C_{id(t)} \in \mathcal{C}$, the event relation induced by t, denoted by $E^p = \langle E, p^E, \mathcal{C}^E\rangle$, is a probabilistic relation whose support relation E has only one attribute, $Event$. E and the probability function p^E are defined by the following two generation rules:*

- *Rule 1:* $t_{e_t} \in E$ *and* $p^E(t_{e_t}) = p(t)$*;*
- *Rule 2:* $\forall C_i \in \mathcal{C} \wedge C_i \neq C_{id(t)}, [(\exists t' \in C_i \wedge t' \succ_s t) \Rightarrow (t_{e_{C_i}} \in E)$ *and* $p^E(t_{e_{C_i}}) = \sum_{t' \succ_s t, t' \in C_i} p(t')]$.

[2]Probability is typically supported as a special attribute in DBMS.

No other tuples belong to E. The partition $\mathcal{C}^E$ is defined as the collection of singleton subsets of E.

Except for one special tuple generated by *Rule 1*, each tuple in the induced event relation (generated by *Rule 2*) represents an event e_{C_i} associated with a part $C_i \in \mathcal{C}$. The probability of this event, denoted by $p(t_{e_{C_i}})$, is the probability that e_{C_i} occurs. Given the tuple t, the *event* e_{C_i} is defined as "some tuple from the part C_i has the score higher than the score of t".

The role of the special tuple t_{e_t} and its probability $p(t)$ will become clear in Thm. 4.3. Let us first look at an example of an induced event relation.

Example 6. *Given R^p as in Example 2, we would like to construct the induced event relation $E^p = \langle E, p^E, \mathcal{C}^E\rangle$ for tuple t=(Temp: 15) from C_2. By Rule 1, we have $t_{e_t} \in E$, $p^E(t_{e_t}) = 0.6$. By Rule 2, since $t \in C_2$, we have $t_{e_{C_1}} \in E$ and $p^E(t_{e_{C_1}}) = \sum_{t' \succ_s t, t' \in C_1} p(t') = p((\text{Temp: } 22)) = 0.6$.*

Therefore,

E:

Event
t_{e_t}
$t_{e_{C_1}}$

p^E:

Prob
0.6
0.6

Proposition 4.1. *Any induced event relation is a simple probabilistic relation.*

Evaluating Global-Top$k$ Queries With the help of *induced event relation*, we could reduce Global-Top$k$ in the general case to Global-Top$k$ in simple probabilistic relations.

Lemma 4.1. *Given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$ and an injective scoring function s, for any $t \in R$, $E^p = \langle E, p^E, \mathcal{C}^E\rangle$. Let $Q^p = \langle E - \{t_{e_t}\}, p^E, \mathcal{C}^E - \{\{t_{e_t}\}\}\rangle$. Then, the Global-Topk probability of t satisfies the following:*

$$P^{R^p}_{k,s}(t) = p(t) \cdot (\sum_{\substack{W_e \in pwd(Q^p) \\ |W_e| < k}} p(W_e)).$$

Theorem 4.3. *Given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$ and an injective scoring function s, for any $t \in R^p$, the Global-Topk probability of t equals the Global-Topk probability of t_{e_t} when evaluating top-k in the induced event relation $E^p = \langle E, p^E, \mathcal{C}^E\rangle$ under the injective scoring function $s^E : E \to \mathbb{R}$, $s^E(t_{e_t}) = \frac{1}{2}$ and $s^E(t_{e_{C_i}}) = i$:*

$$P^{R^p}_{k,s}(t) = P^{E^p}_{k,s^E}(t_{e_t}).$$

Proof. *See* Appendix.

Since any induced event relation is simple (Prop. 4.1), Thm. 4.3 illustrates how we can reduce the computation of $P^{R^p}_{k,s}(t)$ in the original probabilistic relation to a top-k computation in a simple probabilistic relation, where we can apply the DP technique in Section 4.1. The complete algorithms are shown below.

Algorithm 2 (IndEx_Topk) Evaluate Global-Top$k$ queries in a General Probabilistic Relation

Require: $R^p = \langle R, p, \mathcal{C}\rangle, k, s$

1: Initialize a fixed cardinality $k+1$ priority queue Ans of $\langle t, prob\rangle$ pairs, which compares pairs on $prob$;
2: **for** $t \in R$ **do**
3: Calculate $P^{R^p}_{k,s}(t)$ using Algorithm 3
4: Add $\langle t, P^{R^p}_{k,s}(t)\rangle$ to Ans;
5: **if** $|Ans| > k$ **then**
6: remove the pair with the smallest $prob$ value from Ans;
7: **end if**
8: **end for**
9: **return** $\{t | \langle t, P^{R^p}_{k,s}(t)\rangle \in Ans\}$;

In Alg. 3, we first find the part $C_{id(t)}$ where t belongs. In Line 2, we initialize the support relation E of the induced event relation by the tuple generated by Rule 1 in Def. 4.1. For any part C_i other than $C_{id(t)}$(Line 3-8), we compute the probability of the event e_{C_i} according to Def. 4.1. Since all the tuples from the same part are exclusive, this probability is the sum of the probabilities of all tuples that qualify in that part. Note that if no tuple from C_i qualifies, this probability is zero. In this case, we do not care whether any tuple from C_i will be in the possible world or not, since it does not have any influence on whether t will be in top-k or not. The corresponding event tuple is therefore excluded from E. By default, any probabilistic database assumes that any tuple not in the support relation is with probability zero. Line 9 uses Alg. 1 to compute $P^{E^p}_{k,s}(t_{e_t})$. Consequently, we retain only the DP related codes in Alg. 1. Note that Alg. 1 requires all tuples be sorted on score, but this is not a problem for us. Since we already know the scoring function s^E, we simply need to organize tuples based on s^E when generating E. No extra sorting is necessary.

Alg. 2 uses Alg. 3 as a subroutine and computes $P^{R^p}_{k,s}(t)$ for every tuple $t \in R$, then again uses a priority queue to select the final answer set.

4.4. Complexity

For simple probabilistic relations, in Alg. 1, the DP computation takes $O(kn)$ time and using a priority queue to maintain the k highest values takes $O(nlogk)$. So altogether, Alg. 1 takes $O(kn)$. The TA optimization will reduce the computation time on average, however the algorithm will still have the same complexity.

For general probabilistic relations, in Alg. 3, Line 3-8 takes $O(n)$ to build E (we need to scan all tuples within each part). Line 9 uses DP in Alg. 1, which takes $O(|E|k)$, where $|E|$ is no more than the number of parts in partition $\mathcal{C}$, which is in turn no more than n. So Alg. 3 takes $O(kn)$. Alg. 2 repeats Alg. 3 n times, and the priority queue again takes $O(n \log k)$. Altogether, the complexity is $O(kn^2 + n\log k) = O(kn^2)$.

Algorithm 3 (IndEx_Topk_Sub) Calculate $P^{R^p}_{k,s}(t)$using induced event relation

Require: $R^p = \langle R, p, \mathcal{C}\rangle, k, s, t \in R$

1: Find the part $C_{id(t)} \in \mathcal{C}$ such that $t \in C_{id(t)}$;
2: Initialize E with tuple t_{e_t}, where $p^E(t_{e_t}) = p(t)$
$$E = \{t_{e_t}\};$$
3: **for** $C_i \in \mathcal{C}$ and $C_i \neq C_{id(t)}$ **do**
4: $$p(e_{C_i}) = \sum_{\substack{t' \in C_i \\ t' \succ_s t}} p(t');$$
5: **if** $p(e_{C_i}) > 0$ **then**
6: Add a tuple $t_{e_{C_i}}$ to E, where $p^E(t_{e_{C_i}}) = p(e_{C_i})$
$$E = E \cup \{t_{e_{C_i}}\} \ ;$$
7: **end if**
8: **end for**
9: Use Line $2-13$ of Algorithm 1 to compute $P^{E^p}_{k,s^E}(t_{e_t})$;
10: $P^{R^p}_{k,s}(t) = P^{E^p}_{k,s^E}(t_{e_t})$;
11: **return** $P^{R^p}_{k,s}(t)$;

In [20], both U-Top$k$ and U-kRanks take $O(kn)$ in simple probabilistic relations, which is the same as Alg. 1. In the general case, U-Top$k$ takes $\Theta(kmn^{k-1}\log n)$ and U-kRanks takes $\Omega(mn^{k-1})$, where m is a *rule engine* factor. Both U-Top$k$ and U-kRanks do not scale well with k, for the time complexity is already at least cubic when $k \geq 4$. A detailed analysis is available in Appendix.

5. Conclusion

We study the semantic and computational problems for top-k queries in probabilistic databases. We propose three desired properties for a top-k semantics, namely *Exact k*, *Faithfulness* and *Stability*. Our Global-Top$k$ semantics satisfies all of them. We study the computational problem of query evaluation under Global-Top$k$ semantics for simple and general probabilistic relations. For the former case, we propose a dynamic programming algorithm and effectively optimize it with Threshold Algorithm. In the latter case, we show a polynomial reduction to the simple case. In contrast to Soliman et al. [20], our approach satisfies intuitive semantic properties and can be implemented more efficiently. However, [20] is of a more general model as it allows arbitrary tuple generation rules.

6. Future Work

We note that the two dimensions of top-k queries in probabilistic databases, *score* and *probability*, are not treated equally: score is considered in an ordinal sense while probability is considered in a cardinal sense. One of

the future directions would be to integrate *strength of preference* expressed by score into our framework. Another direction is to consider *non-injective* scoring function. A preliminary study shows that this case is non-trivial, because it is not clear how to allocate the probability of a single possible world to different top-k answer sets. Other possible directions include top-k evaluation in other uncertain database models proposed in the literature [2] and more general preference queries in probabilistic databases.

7. Appendix

7.1. Proofs of Table 1

Semantics	Exact k	Faithfulness	Stability
Global-Top$k$	✓(1)	✓(4)	✓(7)
U-Top$k$	× (2)	✓/× (5)	✓(8)
U-kRanks	× (3)	× (6)	× (9)

Table 1. Property satisfaction for different semantics

Proof. The following proofs correspond to the numbers next to each entry in the above table.

Assume that we are given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$, a non-negative integer k and an injective scoring function s.

(1) Global-Top$k$ satisfies *Exact* k.

We compute the Global-Top$k$ probability for each tuple in R. If there is at least k tuples in R, we are always able to pick the k tuples with the highest Global-Top$k$ probability. In case when there are more than $k-r+1$ tuple(s) with the rth highest Global-Top$k$ probability, where $r = 1, 2 \ldots, k$, only $k - r + 1$ of them will be picked nondeterministically.

(2) U-Top$k$ violates *Exact* k.

Example 4 illustrates a counterexample in a simple probabilistic relation.

(3) U-kRanks violates *Exact* k.

Example 4 illustrates a counterexample in a simple probabilistic relation.

(4) Global-Top$k$ satisfies *Faithfulness*.

By the assumption, $t_1 \succ_s t_2$ and $p(t_1) > p(t_2)$, so we need to show that $P_{k,s}(t_1) > P_{k,s}(t_2)$.

For every $W \in pwd(R^p)$ such that $t_2 \in top_{k,s}(W)$ and $t_1 \notin top_{k,s}(W)$, obviously $t_1 \notin W$. Otherwise, since $t_1 \succ_s t_2$, t_1 would be in $top_{k,s}(W)$. Define a world $W' = (W \backslash \{t_2\}) \cup \{t_1\}$, since t_1 and t_2 are either independent or exclusive, $W' \in pwd(R^p)$ and $Pr(W') = Pr(W)\frac{p(t_1)\bar{p}(t_2)}{\bar{p}(t_1)p(t_2)}$. Since $p(t_1) > p(t_2)$, $Pr(W') > Pr(W)$. Moreover, t_1 will substitute for t_2 in the top-k answer to W'.

For the Global-Top$k$ probability of t_1 and t_2, we have

$$\begin{aligned}
P_{k,s}(t_2) &= \sum_{\substack{W\in pwd(R^p)\\ t_1\in top_{k,s}(W)\\ t_2\in top_{k,s}(W)}} Pr(W) + \sum_{\substack{W\in pwd(R^p)\\ t_1\notin top_{k,s}(W)\\ t_2\in top_{k,s}(W)}} Pr(W) \\
&< \sum_{\substack{W\in pwd(R^p)\\ t_1\in top_{k,s}(W)\\ t_2\in top_{k,s}(W)}} Pr(W) + \sum_{\substack{W'\in pwd(R^p)\\ t_1\in top_{k,s}(W')\\ t_2\notin W'}} Pr(W') \\
&\leq \sum_{\substack{W\in pwd(R^p)\\ t_1\in top_{k,s}(W)\\ t_2\in top_{k,s}(W)}} Pr(W) + \sum_{\substack{W'\in pwd(R^p)\\ t_1\in top_{k,s}(W')\\ t_2\notin W'}} Pr(W') \\
&\quad + \sum_{\substack{W''\in pwd(R^p)\\ t_1\in top_{k,s}(W'')\\ t_2\in W''\\ t_2\notin top_{k,s}(W'')}} Pr(W'') \\
&= P_{k,s}(t_1).
\end{aligned}$$

The equality in $\leq$ holds when t_1 and t_2 are exclusive or $s(t_2)$ is among the k highest scores. Since there is at least one inequality in the above equation, we have

$$P_{k,s}(t_1) > P_{k,s}(t_2).$$

(5) U-Top$k$ satisfies *Faithfulness* in simple probabilistic relations while it violates *Faithfulness* in general probabilistic relations.

Simple Probabilistic Relations

Proof. By contradiction. If U-Top$k$ violates *Faithfulness* in a simple probabilistic relation, there exists $R^p = \langle R, p, \mathcal{C}\rangle$ and exists $t_i, t_j \in R, t_i \succ_s t_j, p(t_i) > p(t_j)$, and by U-Top$k$, t_j is in the top-k answer to R^p under the scoring function s while t_i is not.

S is a top-k answer to R^p under the function s by the U-Top$k$ semantics, $t_j \in S$ and $t_i \notin S$. Denote by $Q_{k,s}(S)$ the probability of S under the U-Top$k$ semantics. That is,

$$Q_{k,s}(S) = \sum_{\substack{W\in pwd(R^p)\\ S=top_{k,s}(W)}} Pr(W).$$

For any world W contributing to $Q_{k,s}(S)$, $t_i \notin W$. Otherwise, since $t_i \succ_s t_j$, t_i would be in $top_{k,s}(W)$, which is S. Define a world $W' = (W \backslash \{t_j\}) \cup \{t_i\}$. Since t_i is independent of any other tuple in R, $W' \in$

$pwd(R^p)$ and $Pr(W') = Pr(W)\frac{p(t_i)\bar{p}(t_j)}{\bar{p}(t_i)p(t_j)}$. Moreover, $top_{k,s}(W') = (S\backslash\{t_j\}) \cup \{t_i\}$. Let $S' = (S\backslash\{t_j\}) \cup \{t_i\}$, then W' contributes to $Q_{k,s}(S')$.

$$\begin{aligned}
Q_{k,s}(S') &= \sum_{\substack{W\in pwd(R^p)\\ S'=top_{k,s}(W)}} Pr(W) \\
&\geq \sum_{\substack{W\in pwd(R^p)\\ S=top_{k,s}(W)}} Pr((W\backslash\{t_j\}) \cup \{t_i\}) \\
&= \sum_{\substack{W\in pwd(R^p)\\ S=top_{k,s}(W)}} Pr(W)\frac{p(t_i)\bar{p}(t_j)}{\bar{p}(t_i)p(t_j)} \\
&= \frac{p(t_i)\bar{p}(t_j)}{\bar{p}(t_i)p(t_j)} \sum_{\substack{W\in pwd(R^p)\\ S=top_{k,s}(W)}} Pr(W) \\
&= \frac{p(t_i)\bar{p}(t_j)}{\bar{p}(t_i)p(t_j)} Q_{k,s}(S) \\
&> Q_{k,s}(S),
\end{aligned}$$

which is a contradiction.

General Probabilistic Relations

The following is a counterexample.

Say $k = 2$, $R = \{t_1, t_2, t_3, t_4\}$, $t_1 \succ_s t_2 \succ_s t_3 \succ_s t_4$, t_1 and t_2 are exclusive, t_3 and t_4 are exclusive. $p(t_1) = 0.5, p(t_2) = 0.45, p(t_3) = 0.4, p(t_4) = 0.3$.

By U-Top$k$, the top-2 answer is $\{t_1, t_3\}$, while $t_2 \succ_s t_3$ and $p(t_2) > p(t_3)$, which violates *Faithfulness*.

(6) U-kRanks violates *Faithfulness*.

The following is a counterexample.

Say $k = 2$, R^p is simple. $R = \{t_1, t_2, t_3\}$, $t_1 \succ_s t_2 \succ_s t_3, p(t_1) = 0.48, p(t_2) = 0.8, p(t_3) = 0.78$.

The probabilities of each tuple at each rank are as follows:

	t_1	t_2	t_3
rank 1	0.48	0.416	0.08112
rank 2	0	0.384	0.39936
rank 3	0	0	0.29952

By U-kRanks, the top-2 answer set is $\{t_1, t_3\}$ while $t_2 \succ t_3$ and $p(t_2) > p(t_3)$, which contradicts *Faithfulness*.

(7) Global-Top$k$ satisfies *Stability*.

Proof. In the rest of this proof, let A be the set of all winners under the Global-Top$k$ semantics.

Part I: Probability.

Case 1: Winners.

For any winner $t \in A$, if we only raise the probability of t, we have a new probabilistic relation $(R^p)' = \langle R, p', \mathcal{C}\rangle$, where the new probability function p' is such that $p'(t) > p(t)$ and for any $t' \in R, t' \neq t, p'(t') = p(t')$. Note that $pwd(R^p) = pwd((R^p)')$. In addition, assume $t \in C_t$, where $C_t \in \mathcal{C}$. By Global-Top$k$,

$$P_{k,s}^{R^p}(t) = \sum_{\substack{W\in pwd(R^p)\\ t\in top_{k,s}(W)}} Pr(W)$$

and

$$\begin{aligned}
P_{k,s}^{(R^p)'}(t) &= \sum_{\substack{W\in pwd(R^p)\\ t\in top_{k,s}(W)}} Pr(W)\frac{p'(t)}{p(t)} \\
&= \frac{p'(t)}{p(t)} P_{k,s}^{R^p}(t).
\end{aligned}$$

For any other tuple $t' \in R, t' \neq t$, we have the following equation:

$$\begin{aligned}
P_{k,s}^{(R^p)'}(t') &= \sum_{\substack{W\in pwd(R^p)\\ t'\in top_{k,s}(W), t\in W}} Pr(W)\frac{p'(t)}{p(t)} \\
&\quad + \sum_{\substack{W\in pwd(R^p)\\ t'\in top_{k,s}(W),\ t\notin W\\ (C_t\backslash\{t\})\cap W=\emptyset}} Pr(W)\frac{c-p'(t)}{c-p(t)} \\
&\quad + \sum_{\substack{W\in pwd(R^p)\\ t'\in top_{k,s}(W),\ t\notin W\\ (C_t\backslash\{t\})\cap W\neq\emptyset}} Pr(W) \\
&\leq \frac{p'(t)}{p(t)}\Big(\sum_{\substack{W\in pwd(R^p)\\ t'\in top_{k,s}(W)\\ t\in W}} Pr(W) \\
&\quad + \sum_{\substack{W\in pwd(R^p)\\ t'\in top_{k,s}(W),\ t\notin W\\ (C_t\backslash\{t\})\cap W=\emptyset}} Pr(W) \\
&\quad + \sum_{\substack{W\in pwd(R^p)\\ t'\in top_{k,s}(W),\ t\notin W\\ (C_t\backslash\{t\})\cap W\neq\emptyset}} Pr(W)\Big) \\
&= \frac{p'(t)}{p(t)} P_{k,s}^{R^p}(t'),
\end{aligned}$$

where $c = 1 - \sum_{t''\in C_t\backslash\{t\}} p(t'')$.

Now we can see that, t's Global-Top$k$ probability in $(R^p)'$ will be raised to *exactly* $\frac{p'(t)}{p(t)}$ times of that in R^p

under the same scoring function s, and for any tuple other than t, its Global-Top$k$ probability in $(R^p)'$ can be raised to *as much as* $\frac{p'(t)}{p(t)}$ times of that in R^p under the same scoring function s. As a result, $P_{k,s}^{(R^p)'}(t)$ is still among the highest k Global-Top$k$ probabilities in $(R^p)'$ under the function s, and therefore still a winner.

Case 2: Losers.

This case is similar to *Case 1*.

Part II: Score.

Case 1: Winners.

For any winner $t \in A$, we evaluate R^p under a new scoring function s'. Comparing to s, s' only raises the score of t. That is, $s'(t) > s(t)$ and for any $t' \in R, t' \neq t, s'(t') = s(t')$. Then, in addition to all the worlds already contributing to t's Global-Top$k$ probability when evaluating R^p under s, some other worlds may now contribute to t's Global-Top$k$ probability. Because, under the function s', t might climb high enough to be in the top-k answer set of those worlds.

For any tuple other than t in R, its Global-Top$k$ probability under the function s' either stays the same (if the "climbing" of t does not knock that tuple out of the top-k answer in some possible world) or decreases (otherwise).

Consequently, t is still a winner when evaluating R^p under the function s'.

Case 2: Losers.

This case is similar to *Case 1*.

(8) U-Top$k$ satisfies *Stability*.

Proof. In the rest of this proof, let A be the set of all winners under U-Top$k$ semantics.

Part I: Probability.

Case 1: Winners.

For any winner $t \in A$, if we only raise the probability of t, we have a new probabilistic relation $(R^p)' = \langle R, p', \mathcal{C} \rangle$, where the new probabilistic function p' is such that $p'(t) > p(t)$ and for any $t' \in R, t' \neq t, p'(t') = p(t')$. In the following discussion, we use superscript to indicate the probability in the context of $(R^p)'$. Note that $pwd(R^p) = pwd((R^p)')$.

Recall that $Q_{k,s}(A_t)$ is the probability of a top-k answer set $A_t \subseteq A$ under U-Top$k$ semantics, where $t \in A_t$. Since $t \in A_t$, $Q'_{k,s}(A_t) = Q_{k,s}(A_t)\frac{p'(t)}{p(t)}$.

For any candidate top-k set B other than A_t, i.e. $\exists W \in pwd(R^p), top_{k,s}(W) = B$ and $B \neq A_t$. By definition,

$$Q_{k,s}(B) \leq Q_{k,s}(A_t).$$

For any world W contributing to $Q_{k,s}(B)$, its probability either increase $\frac{p'(t)}{p(t)}$ times (if $t \in W$), or stays the same (if $t \notin W$ and $\exists t' \in W, t'$ and t are exclusive), or decreases (otherwise). Therefore,

$$Q'_{k,s}(B) \leq Q_{k,s}(B)\frac{p'(t)}{p(t)}.$$

Altogether,

$$Q'_{k,s}(B) \leq Q_{k,s}(B)\frac{p'(t)}{p(t)} \leq Q_{k,s}(A_t)\frac{p'(t)}{p(t)} = Q'_{k,s}(A_t).$$

Therefore, A_t is still a top-k answer to $(R^p)'$ under the function s and $t \in A_t$ is still a winner.

Case 2: Losers.

It is more complicated in the case of losers. We need to show that for any loser t, if we decrease its probability, no top-k candidate set B_t containing t will be a new top-k answer set under the U-Top$k$ semantics. The procedure is similar to that in *Case 1*, except that when we analyze the new probability of any original top-k answer set A_i, we need to differentiate between two cases:

(a) t is exclusive with some tuple in A_i;

(b) t is independent of all the tuples in A_i.

It is easier with (a), where all the worlds contributing to the probability of A_i do not contain t. In (b), some worlds contributing to the probability of A_i contain t, while others do not. And we calculate the new probability for those two kinds of worlds differently. As we will see shortly, the probability of A_i stays unchanged in either (a) or (b).

For any loser $t \in R, t \notin A$, by applying the technique used in *Case 1*, we have a new probabilistic relation $(R^p)' = \langle R, p', \mathcal{C} \rangle$, where the new probabilistic function p' is such that $p'(t) < p(t)$ and for any $t' \in R, t' \neq t, p'(t') = p(t')$. Again, $pwd(R^p) = pwd((R^p)')$.

For any top-k answer set A_i to R^p under the function s, $A_i \subseteq A$. Denote by S_{A_i} all the possible worlds contributing to $Q_{k,s}(A_i)$. Based on the membership of t, S_{A_i} can be partitioned into two subsets $S^t_{A_i}$ and $S^{\bar{t}}_{A_i}$.

$$S_{A_i} = \{W | W \in pwd(R^p), top_{k,s}(W) = A_i\};$$
$$S_{A_i} = S^t_{A_i} \cup S^{\bar{t}}_{A_i}, S^t_{A_i} \cap S^{\bar{t}}_{A_i} = \emptyset,$$
$$\forall W \in S^t_{A_i}, t \in W \text{ and } \forall W \in S^{\bar{t}}_{A_i}, t \notin W.$$

If t is exclusive with some tuple in A_i, $S^t_{A_i} = \emptyset$. In this case, any world $W \in S^{\bar{t}}_{A_i}$ contains one of t's exclusive tuples, therefore W's probability will not be affected by the change in t's probability. In this case,

$$\begin{aligned} Q'_{k,s}(A_i) &= \sum_{\substack{W \in pwd(R^p) \\ W \in S^{\bar{t}}_{A_i}}} Pr'(W) = \sum_{\substack{W \in pwd(R^p) \\ W \in S^{\bar{t}}_{A_i}}} Pr(W) \\ &= Q_{k,s}(A_i). \end{aligned}$$

Otherwise, t is independent of all the tuples in A_i. In this case,

$$\frac{\sum_{\substack{W \in pwd(R^p) \\ W \in S^t_{A_i}}} Pr(W)}{\sum_{\substack{W \in pwd(R^p) \\ W \in S^{\bar{t}}_{A_i}}} Pr(W)} = \frac{p(t)}{1 - p(t)}$$

and

$$\begin{aligned} Q'_{k,s}(A_i) &= \sum_{\substack{W \in pwd(R^p) \\ W \in S^t_{A_i}}} Pr(W) \frac{p'(t)}{p(t)} \\ &\quad + \sum_{\substack{W \in pwd(R^p) \\ W \in S^{\bar{t}}_{A_i}}} Pr(W) \frac{1 - p'(t)}{1 - p(t)} \\ &= \sum_{\substack{W \in pwd(R^p) \\ W \in S_{A_i}}} Pr(W) \\ &= Q_{k,s}(A_i). \end{aligned}$$

We can see that in both cases, $Q'_{k,s}(A_i) = Q_{k,s}(A_i)$.

Now for any top-k candidate set containing t, say B_t such that $B_t \not\subseteq A$, by definition, $Q_{k,s}(B_t) < Q_{k,s}(A_i)$. Moreover,

$$Q'_{k,s}(B_t) = Q_{k,s}(B_t) \frac{p'(t)}{p(t)} < Q_{k,s}(B_t).$$

Therefore,

$$Q'_{k,s}(B_t) < Q_{k,s}(B_t) < Q_{k,s}(A_i) = Q'_{k,s}(A_i).$$

Consequently, B_t is still not a top-k answer to $(R^p)'$ under the function s. Since no top-k candidate set containing t can be a top-k answer set to $(R^p)'$ under the function s, t is still a loser.

Part II: Score.

Again, $A_i \subseteq A$ is a top-k answer set to R^p under the function s by U-Top$k$ semantics.

Case 1: Winners.

For any winner $t \in A_i$, we evaluate R^p under a new scoring function s'. Comparing to s, s' only raises the score of t. That is, $s'(t) > s(t)$ and for any $t' \in R, t' \neq t, s'(t') = s(t')$. In some possible world such that $W \in pwd(R^p)$ and $top_{k,s}(W) \neq A_i$, t might climb high enough to be in $top_{k,s'}(W)$. Define T to the set of such top-k candidate sets.

$$\begin{aligned} T = \{&top_{k,s'}(W) | \\ &W \in pwd(R^p), t \notin top_{k,s}(W) \wedge t \in top_{k,s'}(W)\}. \end{aligned}$$

Only a top-k candidate set $B_j \in T$ can possibly end up with a probability higher than that of A_i across all possible worlds, and thus substitute for A_i as a new top-k answer set to R^p under the function s'. In that case, $t \in B_j$, so t is still a winner.

Case 2: Losers.

For any loser $t \in R, t \notin A$. Using a similar technique to *Case 1*, the new scoring function s' is such that $s'(t) < s(t)$ and for any $t' \in R, t' \neq t, s'(t') = s(t')$. When evaluating R^p under the function s', for any world $W \in pwd(R^p)$ such that $t \notin top_{k,s}(W)$, the score decrease of t will not effect its top-k answer, i.e. $top_{k,s'}(W) = top_{k,s}(W)$. For any world $W \in pwd(R^p)$ such that $t \in top_{k,s}(W)$, t might go down enough to drop out of $top_{k,s'}(W)$. In this case, W will contribute its probability to a top-k candidate set without t, instead of the original one with t. In other words, under the function s', comparing to the evaluation under the function s, the probability of a top-k candidate set with t is non-increasing, while the probability of a top-k candidate set without t is non-decreasing[3].

Since any top-k answer set to R^p under the function s does not contain t, it follows from the above analysis that any top-k candidate set containing t will not be a top-k answer set to R^p under the new function s', and thus t is still a loser.

(9) U-kRanks violates *Stability*.

The following is a counterexample.

Say $k = 2$, R^p is simple. $R = \{t_1, t_2, t_3\}$, $t_1 \succ_s t_2 \succ_s t_3$. $p(t_1) = 0.3, p(t_2) = 0.4, p(t_3) = 0.3$.

	t_1	t_2	t_3
rank 1	0.3	0.28	0.126
rank 2	0	0.12	0.138
rank 3	0	0	0.036

By U-kRanks, the top-2 answer set is $\{t_1, t_3\}$.

Now raise the score of t_3 such that $t_1 \succ_{s'} t_3 \succ_{s'} t_2$.

[3]Here, any subset of R with cardinality at most k that is not a top-k candidate set under the function s is conceptually regarded as a top-k candidate set with probability zero under the function s.

	t_1	t_3	t_2
rank 1	0.3	0.21	0.196
rank 2	0	0.09	0.168
rank 3	0	0	0.036

By U-kRanks, the top-2 answer set is $\{t_1, t_2\}$. By raising the score of t_3, we actually turn the winner t_3 to a loser, which contradicts *Stability*.

□

7.2. Proof of Thm. 4.1

Theorem 4.1. *Given a simple probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$ and an injective scoring function s over R^p, if $R = \{t_1, t_2, \ldots, t_n\}$ and $t_1 \succ_s t_2 \succ_s \ldots \succ_s t_n$, the following recursion on Global-Top$k$ queries holds.*

$$q(k,i) = \begin{cases} 0 & k = 0 \\ p(t_i) & 1 \le i \le k \\ (q(k, i-1)\frac{\bar{p}(t_{i-1})}{p(t_{i-1})} + q(k-1, i-1))p(t_i) & \\ & \textit{otherwise} \end{cases}$$

where $q(k,i) = P_{k,s}(t_i)$ and $\bar{p}(t_{i-1}) = 1 - p(t_{i-1})$.

Proof. By induction on k and i.

- Base case.
  - $k = 0$
    For any $W \in pwd(R^p)$, $top_{0,s}(W) = \emptyset$. Therefore, for any $t_i \in R$, the Global-Top$k$ probability of t_i is 0.
  - $k > 0$ and $i = 1$
    t_1 has the highest score among all tuples in R. As long as tuple t_1 appears in a possible world W, it will be in the $top_{k,s}(W)$. So the Global-Top$k$ probability of t_i is the probability that t_1 appears in possible worlds, i.e. $q(k,1) = p(t_1)$.
- Inductive step.
  Assume the theorem holds for $0 \le k \le k_0$ and $1 \le i \le i_0$. For any $W \in pwd(R^p)$, $t_{i_0} \in top_{k_0,s}(W)$ iff $t_{i_0} \in W$ and there are at most $k_0 - 1$ tuples with higher score in W. Note that any tuple with score lower than the score of t_{i_0} does not have any influence on $q(k_0, i_0)$, because its presence/absence in a possible world will not affect the presence of t_{i_0} in the top-k answer of that world.
  Since all the tuples are independent,

$$q(k_0, i_0) = p(t_{i_0}) \cdot \sum_{\substack{W \in pwd(R^p) \\ |\{t | t \in W \wedge t \succ_s t_{i_0}\}| < k_0}} Pr(W).$$

(1) $q(k_0, i_0+1)$ is the Global-Topk_0 probability of tuple t_{i_0+1}.

$$\begin{aligned} q(k_0, i_0+1) &= \sum_{\substack{W \in pwd(R^p) \\ t_{i_0+1} \in top_{k_0,s}(W) \\ t_{i_0} \in top_{k_0,s}(W)}} Pr(W) \\ &+ \sum_{\substack{W \in pwd(R^p) \\ t_{i_0+1} \in top_{k_0,s}(W) \\ t_{i_0} \in W,\ t_{i_0} \notin top_{k_0,s}(W)}} Pr(W) \\ &+ \sum_{\substack{W \in pwd(R^p) \\ t_{i_0+1} \in top_{k_0,s}(W) \\ t_{i_0} \notin W}} Pr(W). \end{aligned}$$

For the first part of the left hand side,

$$\sum_{\substack{W \in pwd(R^p) \\ t_{i_0+1} \in top_{k_0,s}(W) \\ t_{i_0} \in top_{k_0-1,s}(W)}} Pr(W) = p(t_{i_0+1}) q(k_0 - 1, i_0).$$

The second part is zero. Since $t_{i_0} \succ_s t_{i_0+1}$, if $t_{i_0+1} \in top_{k_0,s}(W)$ and $t_{i_0} \in W$, then $t_{i_0} \in top_{k_0,s}(W)$.
The third part is the sum of the probabilities of all possible worlds such that $t_{i_0+1} \in W, t_{i_0} \notin W$ and there are at most $k_0 - 1$ tuples with score higher than the score of t_{i_0} in W. So it is equivalent to

$$\begin{aligned} & p(t_{i_0+1})\bar{p}(t_{i_0}) \cdot \sum_{|\{t | t \in W \wedge t \succ_s t_{i_0}\}| < k_0} Pr(W) \\ = \ & p(t_{i_0+1})\bar{p}(t_{i_0}) \frac{q(k_0, i_0)}{p(t_{i_0})}. \end{aligned}$$

Altogehter, we have

$$\begin{aligned} & q(k_0, i_0+1) \\ = \ & p(t_{i_0+1}) q(k_0 - 1, i_0) + p(t_{i_0+1})\bar{p}(t_{i_0}) \tfrac{q(k_0,i_0)}{p(t_{i_0})} \\ = \ & (q(k_0 - 1, i_0) + q(k_0, i_0)\tfrac{\bar{p}(t_{i_0})}{p(t_{i_0})}) p(t_{i_0+1}). \end{aligned}$$

(2) $q(k_0+1, i_0)$ is the Global-Top(k_0+1) probability of tuple t_{i_0}. Use a similar argument as above, it can be shown that this case is correctly computed by Eqn. (2) as well. □

7.3. Proof for Thm. 4.2

Theorem 4.2 (Correctness)**.** *Given a simple probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$, a non-negative integer k and an injective scoring function s over R^p, the above TA-based algorithm correctly finds a top-k answer under Global-Top$k$ semantics.*

Proof. In every iteration of Step (2), say $\underline{t} = t_i$, for any unseen tuple t, s' is an injective scoring function over R^p, which only differs from s in the score of t. Under the function s', $t_i \succ_{s'} t \succ_{s'} t_{i+1}$. If we evaluate the top-k query in R^p under s' instead of s, $P_{k,s'}(t) = \frac{p(t)}{\underline{p}} UP$. On the other hand, for any $W \in pwd(R^p)$, W contributing to $P_{k,s}(t)$ implies that W contributes to $P_{k,s'}(t)$, while the reverse is not necessarily true. So, we have $P_{k,s'}(t) \geq P_{k,s}(t)$. Recall that $\underline{p} \geq p(t)$, therefore $UP \geq \frac{p(t)}{\underline{p}} UP = P_{k,s'}(t) \geq P_{k,s}(t)$. The conclusion follows from the correctness of the original TA algorithm and Alg. 1. □

7.4. Proof for Lemma 4.1

Lemma 4.1. *Given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$ and an injective scoring function s, for any $t \in R$, $E^p = \langle E, p^E, \mathcal{C}^E\rangle$. Let $Q^p = \langle E - \{t_{e_t}\}, p^E, \mathcal{C}^E - \{\{t_{e_t}\}\}\rangle$. Then, the Global-Topk probability of t satisfies the following:*

$$P^{R^p}_{k,s}(t) = p(t) \cdot \Big(\sum_{\substack{W_e \in pwd(Q^p) \\ |W_e| < k}} p(W_e) \Big).$$

Proof. Given $t \in R$, k and s, let A be a subset of $pwd(R^p)$ such that $W \in A \Leftrightarrow t \in top_{k,s}(W)$. If we group all the possible worlds in A by the set of parts whose tuple in W has higher score than the score of t, then we will have the following partition:

$$A = A_1 \cup A_2 \cup \ldots \cup A_q, A_i \cap A_j = \emptyset, i \neq j$$

and

$$\forall A_i, \forall W_1, W_2 \in A_i, i = 1, 2, \ldots, q,$$
$$\{C_j | \exists t' \in W_1 \cap C_j, t' \succ_s t\} = \{C_j | \exists t' \in W_2 \cap C_j, t' \succ_s t\}.$$

Moreover, denote $CharParts(A_i)$ to A_i's characteristic set of parts.

Now, let B be a subset of $pwd(E^p)$, such that $W_e \in B \Leftrightarrow |W_e| < k$. There is a bijection $g : \{A_i | A_i \subseteq A\} \rightarrow B$, mapping each part A_i in A to the a possible world in B which contains only tuples belonging to parts in A_i 's characteristic set.

$$g(A_i) = \{t_{e_{C_j}} | C_j \in CharParts(A_i)\}.$$

The following equation holds from the definition of induced event relation and Prop. 4.1.

$$\begin{aligned} \sum_{W \in A_i} Pr(W) &= p(t) \cdot \prod_{C_i \in CharParts(A_i)} p(t_{e_{C_i}}) \\ &= p(t) Pr(g(A_i)). \end{aligned}$$

Therefore,

$$\begin{aligned} P^{R^p}_{k,s}(t) &= \sum_{W \in A} Pr(W) = \sum_{i=1}^{q} \Big(\sum_{W \in A_i} Pr(W) \Big) \\ &= \sum_{i=1}^{q} p(t) Pr(g(A_i)) = p(t) \sum_{i=1}^{q} Pr(g(A_i)) \\ &= p(t) \sum_{W_e \in B} Pr(W_e). \end{aligned}$$

□

7.5. Proof for Thm. 4.3

Theorem 4.3. *Given a probabilistic relation $R^p = \langle R, p, \mathcal{C}\rangle$ and an injective scoring function s, for any $t \in R^p$, the Global-Topk probability of t equals the Global-Topk probability of t_{e_t} when evaluating top-k in the induced event relation $E^p = \langle E, p^E, \mathcal{C}^E\rangle$ under the injective scoring function $s^E : E \rightarrow \mathbb{R}$, $s^E(t_{e_t}) = \frac{1}{2}$ and $s^E(t_{e_{C_i}}) = i$:*

$$P^{R^p}_{k,s}(t) = P^{E^p}_{k,s^E}(t_{e_t}).$$

Proof. Since t_{e_t} has the lowest score under s^E, for any $W_e \in pwd(E^p)$, the only chance $t_{e_t} \in top_{k,s^E}(W_e)$ is when there are at most k tuples in W_e, including t_{e_t}.

$$\forall W_e \in pwd(E^p),$$
$$t_{e_t} \in top_{k,s}(W_e) \Leftrightarrow (t_{e_t} \in W_e \wedge |W_e| \leq k).$$

Therefore,

$$P^{E^p}_{k,s^E}(t_{e_t}) = \sum_{t_{e_t} \in W_e \wedge |W_e| \leq k} Pr(W_e).$$

In the proof of Lemma 4.1, B contains all the possible worlds having at most $k-1$ tuples from $E - \{t_{e_t}\}$. By Prop. 4.1,

$$\sum_{t_{e_t} \in W_e \wedge |W_e| \leq k} Pr(W_e) = p(t) \sum_{W'_e \in B} Pr(W'_e).$$

By Lemma 4.1,

$$p(t) \sum_{W'_e \in B} Pr(W'_e) = P^{R^p}_{k,s}(t).$$

Consequently,

$$P^{R^p}_{k,s}(t) = P^{E^p}_{k,s^E}(t_{e_t}).$$

□

7.6. Complexity Analysis on U-Top$k$ and U-kRanks

7.6.1 U-Top$k$

We study the OptU-Top$k$ algorithm proposed in [20] for U-Top$k$. First of all, it is worth mentioning that in OptU-Top$k$, the notation $s_{l,i}$ does not uniquely denote a state. It can be any state such that it is

- a top-l tuple vector in one or more possible worlds;

- t_i is the tuple last seen in the score-ranked stream for that state.

For example, we have seen t_1 and t_2 in the score-ranked stream. $s_{1,2}$ could be $\langle t_1, \neg t_2\rangle$ or $\langle \neg t_1, t_2\rangle$.

In the following discussion, we are going to use superscript to refer to a unique state. The superscript is a bit vector indicating the membership of tuples seen so far for that state. For example, $s^{10}_{1,2} = \langle t_1, \neg t_2\rangle$ and $s^{01}_{1,2} = \langle \neg t_1, t_2\rangle$.

Now, assume n is the number of all tuples, consider an example satisfying the following two conditions:

- *Condition 1:* all the tuples are of positive probability less than $\frac{1}{2}$, i.e. $\forall t, 0 < p(t) < \frac{1}{2}$;

- *Condition 2:* state $s^{b1}_{k,n}$ is the winner under OptU-Top$k$. Note that the bit vector ends with 1, which means the last tuple seen, i.e. t_n, is in $s^{b1}_{k,n}$. In other words, we need to exhaust the source to find the final answer.

In this case, because all the tuples have probability less than $\frac{1}{2}$, for all $l' < l$ and all possible bit vectors b' for state $s^{b'}_{l',i}$

$$Pr(s^{b'}_{l',i}) > Pr(s^{b}_{l,i}) \tag{3}$$

That is, the probability of $s^b_{l,i}$ is less than the probability of any state that has seen the same tuples but is of lower length.

On the other hand, we have

$$Pr(s^{b}_{l-1,i-1}) > Pr(s^{b0}_{l-1,i}) > Pr(s^{b1}_{l,i}) \tag{4}$$

Eqn. (4) says the probability of $s^b_{l-1,i-1}$ is higher than the probability of any state derived directly from it (Property 1 in [20]). The second inequality in Eqn. (4) follows from Eqn. (3).

By *Condition 2* above, $s^{b1}_{k,n}$ is a winner. By applying Eqn. (3) (4) recursively, in the worst case, it will enumerates at least all the states of length $k-1$. Therefore, there can be at least $\sum_{i=1}^{k-1}\binom{n}{i}$ states being generated and inserted into the priority queue before the source is exhausted or an answer has to be reported. That implies that there could be this many **while** loops (Line 5-19) in OptU-Top$k$ algorithm. At the jth iteration, the state extension at Line 16 takes $2m$, where m is a *rule engine* factor. The insertion to the priority queue at Line 17 takes $2\log(j-1)$ for the length $(j-1)$ queue. Altogether, the algorithm costs

$$\begin{aligned} & 4m(\sum_{j=1}^{(\sum_{i=1}^{k-1}\binom{n}{i})} \log j) \\ = \;& 4m(\log(\sum_{i=1}^{k-1}\binom{n}{i})!) \\ = \;& \Theta(m(\sum_{i=1}^{k-1}\binom{n}{i})\log(\sum_{i=1}^{k-1}\binom{n}{i})). \end{aligned}$$

By assuming $k \ll n$, and thus $(\sum_{i=1}^{k-1}\binom{n}{i}) = \Theta(n^{k-1})$, the above equation yields

$$\Theta(kmn^{k-1}\log n).$$

7.6.2 U-kRanks

We study the OptU-kRanks algorithm proposed in [20] for U-kRanks. In the worst case when the source is exhausted, the **while** loop (Line 6-25) will run n times. The **for** loop (Line 8-23) runs for k times when $depth > k$. For each **for** loop, in Line 9, there could be $\binom{depth}{i-1}$ many states, each needs the rule engine factor m for the extension. Altogether, the algorithm costs

$$m\sum_{depth=1}^{n}(\sum_{i=1}^{k}\binom{depth}{i-1}).$$

By assuming $k \ll depth < n$ in general, and thus $(\sum_{i=1}^{k}\binom{depth}{i-1}) = \Theta(depth^{k-1})$, the above equation yields

$$\Omega(mn^{k-1}).$$

7.6.3 Simple Probabilistic Relations

For a simple probabilistic relation, where all the tuples are independent, [20] provides efficient optimization algorithms IndepU-Top$k$ and IndepU-kRanks for U-Top$k$ and U-kRanks respectively. IndepU-Top$k$ keeps one state for each length. IndepU-kRanks uses dynamic programming. Both optimization algorithms are of $O(kn)$ time complexity.

7.6.4 Comments

The worst case happens to OptU-Top$k$ and OptU-kRanks when all the tuples are independent but no optimization technique is used. Since [20] has separate optimization for the independent case, the worst case will thus be the "almost" independent case, where most tuples are independent but not all.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases : The Logical Level*. Addison Wesley, 1994.

[2] L. Antova, C. Koch, and D. Olteanu. World-set decompositions: Expressiveness and efficient algorithms. In *ICDT*, 2007.

[3] O. Benjelloun, A. D. Sarma, A. Y. Halevy, and J. Widom. Uldbs: Databases with uncertainty and lineage. In *VLDB*, 2006.

[4] N. Bruno and H. Wang. The threshold algorithm: From middleware systems to the relational engine. *IEEE Trans. Knowl. Data Eng.*, 19(4):523–537, 2007.

[5] R. Cavallo and M. Pittarelli. The theory of probabilistic databases. In *VLDB*, 1987.

[6] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[7] R. Fagin. Combining fuzzy information from multiple systems. *J. Comput. Syst. Sci.*, 58(1):83–99, 1999.

[8] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *PODS*, 2001.

[9] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Syst.*, 15(1):32–66, 1997.

[10] S. Guha, N. Koudas, A. Marathe, and D. Srivastava. Merging the results of approximate match operations. In *VLDB*, pages 636–647, 2004.

[11] J. Y. Halpern. An analysis of first-order logics of probability. *Artif. Intell.*, 46(3):311–350, 1990.

[12] I. F. Ilyas, W. G. Aref, and A. K. Elmagarmid. Joining ranked inputs in practice. In *VLDB*, 2002.

[13] I. F. Ilyas, W. G. Aref, and A. K. Elmagarmid. Supporting top-k join queries in relational databases. In *VLDB*, 2003.

[14] T. Imielinski and W. L. Jr. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984.

[15] L. V. S. Lakshmanan, N. Leone, R. B. Ross, and V. S. Subrahmanian. Probview: A flexible probabilistic database system. *ACM Trans. Database Syst.*, 22(3):419–469, 1997.

[16] A. Marian, N. Bruno, and L. Gravano. Evaluating top- queries over web-accessible databases. *ACM Trans. Database Syst.*, 29(2):319–362, 2004.

[17] http://www.infosys.uni-sb.de/projects/maybms/.

[18] A. Natsev, Y.-C. Chang, J. R. Smith, C.-S. Li, and J. S. Vitter. Supporting incremental join queries on ranked inputs. In *VLDB*, 2001.

[19] C. Ré, N. N. Dalvi, and D. Suciu. Efficient top-k query evaluation on probabilistic data. In *ICDE*, 2007.

[20] M. A. Soliman, I. F. Ilyas, and K. C.-C. Chang. Top-k query processing in uncertain databases. In *ICDE*, 2007.

[21] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *CIDR*, 2005.

[22] X. Zhang and J. Chomicki. On the semantics and evaluation of top-k queries in probabilistic databases. Technical report, Dept. of Comp. Sci. and Engr., University at Buffalo, SUNY, Dec 2007.

[23] E. Zimányi. Query evaluation in probabilistic relational databases. *Theor. Comput. Sci.*, 171(1-2):179–219, 1997.