Consensus Answers for Queries over Probabilistic Databases

Jian Li and Amol Deshpande
{lijian, amol}@cs.umd.edu
University of Maryland at College Park

ABSTRACT

We address the problem of finding a “best” deterministic query answer to a query over a probabilistic database. For this purpose, we propose the notion of a consensus world (or a consensus answer) which is a deterministic world (answer) that minimizes the expected distance to the possible worlds (answers). This problem can be seen as a generalization of the well-studied inconsistent information aggregation problems (e.g. rank aggregation) to probabilistic databases. We consider this problem for various types of queries including SPJ queries, Top-k ranking queries, group-by aggregate queries, and clustering. For different distance metrics, we obtain polynomial time optimal or approximation algorithms for computing the consensus answers (or prove NP-hardness). Most of our results are for a general probabilistic database model, called *and/xor tree model*, which significantly generalizes previous probabilistic database models like x-tuples and block-independent disjoint models, and is of independent interest.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Query Processing

General Terms

Theory, Algorithms

Keywords

Consensus answers, rank aggregation, probabilistic databases, query processing, probabilistic and/xor tree

1. INTRODUCTION

There is an increasing interest in uncertain and probabilistic databases arising in application domains like information retrieval [14, 38], recommendation systems [34, 36], mobile object data management [8], information extraction [23], data integration [3] and sensor networks [15]. Supporting complex queries and decision-making on probabilistic databases is significantly more difficult than on deterministic databases, and the key challenges include defining proper and intuitive semantics for queries over them, and developing efficient query processing algorithms.

The common semantics in probabilistic databases are the *possible worlds semantics*, where a probabilistic database is considered to correspond to a probability distribution over a set of deterministic databases called possible worlds. Therefore, posing queries over such a probabilistic database generates a probability distribution over a set of deterministic results which we call “possible answers”. However, a full list of possible answers together with their probabilities is not desirable in most cases since the size of the list could be exponentially large, and the probability associated with each single answer is extremely small. One approach to addressing this issue is to “combine” the possible answers somehow to obtain a more compact representation of the result. For select-project-join queries, for instance, one proposed approach is to union all the possible answers, and compute the probability of each result tuple by adding the probabilities of all the possible answers it belongs to [14]. This approach, however, can not be easily extended to other types of queries like ranking or aggregate queries.

Furthermore, from the user or application perspective, despite the probabilistic nature of the data, a single, deterministic query result may be desirable, on which further analysis or decision-making could be based. For SPJ queries, this is often achieved by “thresholding”, i.e., returning only the result tuples with a sufficiently high probability of being true. For aggregate queries, often expected values are returned instead [28]. For ranking queries, on the other hand, a range of different approaches have been proposed to find the true ranking of the tuples. These include UTop-k [40], URank-k [40], probabilistic threshold Top-k [26], global Top-k [46], expected rank [10], and so on. Although these definitions seem to reason about the ranking over probabilistic databases in some “natural” ways, there is a lack of a unified and systematic analysis framework to justify their semantics and to discriminate the usefulness of one from another.

In this paper, we consider the problem of combining the results for all possible worlds in a systematic way by putting it in the context of *inconsistent information aggregation* which has been studied extensively in numerous contexts over the last half century. In our context, the set of different query answers returned from possible worlds can be thought as inconsistent information which we need to aggregate to obtain a single representative answer. To the best of our knowledge, this connection between query processing in probabilistic databases and inconsistent information aggregation, though natural, has never been realized before in any formal and mathematical way. Concretely, we propose the notion of *the con-*

[1] This research was supported by NSF under grants CCF-0728839 and IIS-0546136.

PODS’09, June 29–July 2, 2009, Providence, Rhode Island, USA.
Copyright 2009 ACM 978-1-60558-553-6 /09/06 ...$5.00.

sensus answer. Roughly speaking, the consensus answer is a answer that is *closest in expectation* to the answers of the possible worlds. To measure the closeness of two answers τ_1 and τ_2, we need to define suitable distance function $\mathsf{d}(\tau_1, \tau_2)$ over the answer space. For example, if an answer is a vector, we can simply use the L_2 norm; whereas in other cases, for instance, Top-k queries, the definition of d is more involved. If the most consensus answer can be taken from any point in the answer space, we refer it as the *mean answer*. A *median answer*, on the other hand, must be the answer for some possible world with non-zero probability.

From a mathematical perspective, if the distance function is properly defined to reflect the closeness of the answers, the most consensus answer is perhaps the best deterministic representative of the set of all possible answers, since it can be thought as the centroid of the set of points corresponding to the possible answers.

Our key contributions can be summarized as follows:

- (Probabilistic And/Xor Tree) We propose a new model for modeling correlations, called the *probabilistic and/xor tree* model, that can capture two types of correlations, mutual exclusion and coexistence. This model generalizes the previous models such as x-tuples, and block-independent disjoint tuples model. More important, this model admits an elegant generating functions based framework for many types of probability computations. We note that it is possible to represent the correlations captured by such a tree using probabilistic c-tables [22] and provenance semirings [21]. However, that does not directly imply efficient algorithms for the problems we consider in this paper.

- (Set Distance Metrics) We show that the mean and the median world can be found in polynomial time for the *symmetric difference* metric for and/xor tree model. For the Jaccard distance metric, we present a polynomial time algorithm to compute the mean and median world for a tuple independent database.

- (Top-k ranking Queries) The problem of aggregating inconsistent rankings has been well-studied under the name of *rank aggregation* [16]. We develop polynomial time algorithms for computing mean and median top-k answers under the symmetric difference metric, and the mean answers under *intersection metric* and *generalized Spearman's footrule distance* [18], for the and/xor tree model.

- (Groupby Aggregates) For group by count queries, we present a 4-approximation to the problem of finding a median answer (finding mean answers is trivial).

- (Consensus Clustering) We also consider the consensus clustering problem for the and/xor tree model and get a constant approximation by extending a previous result [2].

Outline: We begin with a discussion of the related work (Section 2). We then define the probabilistic and/xor tree model (Section 3), and present a generating functions-based method to do probability computations on them (Section 3.3). The bulk of our key results are presented in Sections 4 and 5 where we address the problem of finding consensus worlds for different set distance metrics and for top-k ranking queries respectively. We then briefly discuss finding consensus worlds for group-by *count* aggregate queries and clustering queries in Section 6.

2. RELATED WORK

There has been much work on managing probabilistic, uncertain, incomplete, and/or fuzzy data in database systems and this area has received renewed attention in the last few years (see e.g. [27, 5, 30, 19, 20, 8, 14, 37, 44, 4, 43]). This work has spanned a range of issues from theoretical development of data models and data languages, to practical implementation issues such as indexing techniques. In terms of representation power, most of this work has either assumed independence between the tuples [19, 14], or has restricted the correlations that can be modeled [5, 30, 3, 37]. Several approaches for modeling complex correlations in probabilistic databases have also been proposed [38, 4, 39, 43].

For efficient query evaluation over probabilistic databases, one of the key results is the dichotomy of conjunctive query evaluation on tuple-independent probabilistic databases by Dalvi and Suciu [14, 13]. Briefly the result states that the complexity of evaluating a conjunctive query over tuple-independent probabilistic databases is either PTIME or #P-complete. For the former case, Dalvi and Suciu [14] also present an algorithm to find what are called *safe query plans*, that permit correct *extensional* evaluation of the query. Unfortunately the problem of finding consensus answers appears to be much harder; this is because even if a query has a safe plan, the result tuples may still be arbitrarily correlated.

In recent years, there has also been much work on efficiently answering different types of queries over probabilistic databases, including aggregates [28], summarization [12], clustering [11], nearest neighbors [6] and so on. Soliman et al. [40] first considered the problem of top-k query evaluation over probabilistic databases, and proposed two ranking functions to combine the tuple scores and probabilities. This problem is particularly interesting for our purposes, since the semantics of the query (what it should return) are not quite clear. This has led to much recent work (Zhang et al. [46], Hua et al. [25, 26], Cormode et al. [10] etc.) that has proposed different ways to compute the top-k answers; as we observe in our recent work, the answers under different semantics can be wildly different from each other [31].

The problem of aggregating inconsistent information from different sources arises in numerous disciplines and has been studied in different contexts over decades. Specifically, the RANK-AGGREGATION problem aims at combining k different complete ranked lists $\tau_1, \ldots, \tau_k$ on the same set of objects into a single ranking, which is the best description of the combined preferences in the given lists. This problem was considered as early as the 18th century when Condorcet and Borda proposed a voting system for elections [9, 7]. In the late 50's, Kemeny proposed the first mathematical criterion for choosing the best ranking [29]. Namely, the Kemeny optimal aggregation τ is the ranking that minimizes $\sum_{i=1}^{k} \mathsf{d}(\tau, \tau_i)$, where $\mathsf{d}(\tau_i, \tau_j)$ is the number of pairs of elements that are ranked in different order in τ_i and τ_j (also called Kendall's tau distance). While computing the Kemeny optimal is shown to be NP-hard [17], 2-approximation can be easily achieved by picking the best ranking from k given ranking lists. The other well-known 2-approximation is from the fact the Spearman footrule distance, defined to be $\mathsf{d}_F(\tau_i, \tau_j) = \sum_t |\tau_i(t) - \tau_j(t)|$, is within twice the Kendall's tau distance and the footrule aggregation can be done optimally in polynomial time [16]. Ailon et al. [2] improve the approximation ratio to $4/3$. We refer the readers to [24] for a survey on this problem. For aggregating top-k answers, Ailon [1] recently obtained an $3/2$-approximation based on rounding an LP solution. Quite recently, Soliman et al. [41] also observed the relationship between ranking in uncertain databases and the RANK-AGGREGATION problem and proposed a polynomial time algorithm under Spearman's footrule distance for full rankings.

The CONSENSUS-CLUSTERING problem asks for the best clustering of a set of elements which minimizes the number of pairwise disagreements with the given k clusterings. It is known to be NP-hard [42] and a 2-approximation can also be obtained by picking the best one from the given k clusterings. The best known approximation ratio is $4/3$ [2].

3. PRELIMINARIES

We begin with reviewing the possible worlds semantics, and introduce the probabilistic and/xor tree model.

3.1 Possible World Semantics

We consider probabilistic databases with both tuple-level uncertainty (the existence of a tuple is uncertain) and attribute-level uncertainty (a tuple attribute value is uncertain). Specifically, we denote a probabilistic relation by $R^P(K; A)$, where K is the *key* attribute, and A is the *value* attribute[1]. For a particular tuple in R^P, its key attribute is certain and is sometimes called the possible worlds key. R^P is assumed to correspond to a probability space (PW, Pr) where the set of outcomes is a set of deterministic relations, which we call *possible worlds*, $PW = \{pw_1, pw_2, ..., pw_N\}$. Note that two tuples can not have the same value for the key attribute in a single possible world. Because of the typically exponential size of PW, an explicit possible worlds representation is not feasible, and hence the semantics are usually captured implicitly by probabilistic models with polynomial size specification.

Let T denote the set of tuples in all possible worlds. For ease of notation, we will use $t \in pw$ in place of "t appears in the possible world pw", $\mathsf{Pr}(t)$ to denote $\mathsf{Pr}(\text{t is present})$ and $\mathsf{Pr}(\neg t)$ to denote $\mathsf{Pr}(\text{t is not present})$.

Further, for a tuple $t^P \in R^P$, we call the certain tuples corresponding to it (with the same key value) in the union of the possible worlds, its *alternatives*.

Block-Independent Disjoint (BID) Scheme: BID is one of the more popular models for probabilistic databases, and assumes that different probabilistic tuples (with different key values) are independent of each other [14, 37, 13, 35]. Formally, a BID scheme has the relational schema of the from $R(K; A; \mathsf{Pr})$ where K is the possible worlds key, A is the value attribute, and Pr captures the probability of the corresponding tuple alternative.

3.2 Probabilistic And/Xor Tree

We generalize the block-independent disjoint tuples model, which can capture *mutual exclusion* between tuples, by adding support for *mutual co-existence*, and allowing these to be specified in a hierarchical manner. Two events satisfy the mutual co-existence correlation if in any possible world, either both happen or neither occurs. We model such correlations using a *probabilistic and/xor tree* (or and/xor tree for short), which also generalizes the notions of *x-tuples* [37, 45], p-or-sets [13] and tuple independent databases. We first considered this model for tuple-level uncertainty in an earlier paper [31], and generalize it here to handle attribute-level uncertainty.

We use $\ovee$ (or) to denote mutual exclusion and $\owedge$ (and) for coexistence. Figure 1 shows two examples of probabilistic and/xor trees. Briefly, the leaves of the tree correspond to the tuple alternatives (we abuse the notation somewhat and use t_i to denote both the tuple, and its key value). The first tree captures a relation with four independent tuples, t_1, t_2, t_3, t_4, each with two alternatives, whereas the second tree shows how we can capture arbitrary possible worlds using an and/xor tree (Figure 1(ii) shows the possible worlds corresponding to that tree).

Now, let us formally define a probabilistic and/xor tree. In tree $\mathcal{T}$, we denote the set of children of node v by $Ch_{\mathcal{T}}(v)$ and the least common ancestor of two leaves l_1 and l_2 by $LCA_{\mathcal{T}}(l_1, l_2)$. We omit the subscript if the context is clear.

DEFINITION 1. *A probabilistic and/xor tree $\mathcal{T}$ represents the mutual exclusion and co-existence correlations in a probabilistic relation $R^P(K; A)$, where K is the possible worlds key, and A is the value attribute. In $\mathcal{T}$, each leaf is a key-attribute pair (a tuple alternative), and each inner node has a mark, $\ovee$ or $\owedge$. For each $\ovee$ node u and each of its children $v \in Ch(u)$, there is a nonnegative value $\mathsf{Pr}(u, v)$ associated with the edge (u, v). Moreover, we require*

- *(Probability Constraint)* $\sum_{v: v \in Ch(u)} \mathsf{Pr}(u, v) \leq 1$.
- *(Key Constraint) For any two different leaves l_1, l_2 holding the same key, $LCA(l_1, l_2)$ is a $\ovee$ node*[2].

Let $\mathcal{T}_v$ be the subtree rooted at v and $Ch(v) = \{v_1, \ldots, v_\ell\}$. The subtree $\mathcal{T}_v$ inductively defines a random subset S_v of its leaves by the following independent process:

- *If v is a leaf, $S_v = \{v\}$.*
- *If $\mathcal{T}_v$ roots at a $\ovee$ node, then*
$$S_v = \begin{cases} S_{v_i} & \textit{with prob } \mathsf{Pr}(v, v_i) \\ \emptyset & \textit{with prob } 1 - \sum_{i=1}^{\ell} \mathsf{Pr}(v, v_i) \end{cases}$$
- *If $\mathcal{T}_v$ roots at a $\owedge$ node, then $S_v = \cup_{i=1}^{\ell} S_{v_i}$*

Probabilistic and/xor trees can capture more complicated correlations than the prior models such as the BID model or x-tuples. We remark that Markov or Bayesian network models are able to capture more general correlations [38], however, the structure of the model is more complex and probability computations on them (inference) is typically exponential in the treewidth of the model. The treewidth of an and/xor tree (viewing it as a Markov network) is not bounded, and hence the techniques developed for those models can not be used to obtain a polynomial time algorithms for and/xor trees.

3.3 Computing Probabilities on And/Xor Trees

Aside from the representational power of the and/xor tree model, perhaps its best feature is that many types of probability computations can be done efficiently and elegantly on them using *generating functions*. In our prior work [31], we used a similar technique for computing ranking functions for tuple-level uncertainty model. Here we generalize the idea to a broader range of probability computations.

We denote the and/xor tree by $\mathcal{T}$. Suppose $\mathcal{X} = \{x_1, x_2, \ldots\}$ is a set of variables. Define a mapping s which associates each leaf $l \in \mathcal{T}$ with a variable $s(l) \in \mathcal{X}$. Let $\mathcal{T}_v$ denote the subtree rooted at v and let $v_1, \ldots, v_l$ be v's children. For each node $v \in \mathcal{T}$, we define a generating function $\mathcal{F}_v$ recursively:

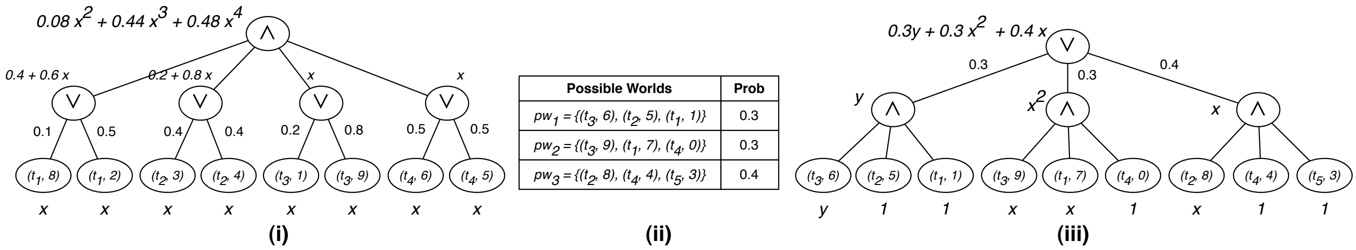
- If v is a leaf, $\mathcal{F}_v^i(\mathcal{X}) = s(v)$.
- If v is a $\ovee$ node,
$$\mathcal{F}_v(\mathcal{X}) = (1 - \sum_{h=1}^{l} p(v, v_h)) + \sum_{h=1}^{l} \mathcal{F}_{v_h}(\mathcal{X}) \cdot p(v, v_h)$$
- If v is a $\owedge$ node, $\mathcal{F}_v^i(\mathcal{X}) = \prod_{h=1}^{l} \mathcal{F}_{v_h}(\mathcal{X})$.

The generating function $\mathcal{F}(\mathcal{X})$ for tree $\mathcal{T}$ is the one defined above for the root. It is easy to see, if we have a constant number of variables, the polynomial can be expanded in the form of $\sum_{i_1, i_2, \ldots} c_{i_1, i_2 \ldots} x_1^{i_1} x_2^{i_2} \ldots$ in polynomial time.

Now recall that each possible world pw contains a subset of the leaves of $\mathcal{T}$ (as dictated by the $\ovee$ and $\owedge$ nodes). The following theorem characterizes the relationship between the coefficients of $\mathcal{F}$ and the probabilities we are interested in.

[1] For clarity, we assume singleton key and value attributes.

[2] The key constraint is imposed to avoid two leaves with the same key but different attribute values coexisting in a possible world.



Possible Worlds	Prob
$pw_1 = \{(t_3, 6), (t_2, 5), (t_1, 1)\}$	0.3
$pw_2 = \{(t_3, 9), (t_1, 7), (t_4, 0)\}$	0.3
$pw_3 = \{(t_2, 8), (t_4, 4), (t_5, 3)\}$	0.4

Figure 1: (i) The and/xor tree representation of a set of block-independent disjoint tuples; the generating function obtained by assigning the same variable x to all leaves gives us the distribution over the sizes of the possible worlds. (ii) Example of a highly correlated probabilistic database with 3 possible worlds and (iii) the and/xor tree that captures the correlation; the coefficient of y (0.3) is $\mathsf{Pr}(r(t_3, 6) = 1)$ (i.e., prob. that that alternative of t_3 is ranked at position 1).

THEOREM 1. *The coefficient of the term $\prod_j x_j^{i_j}$ in $\mathcal{F}(\mathcal{X})$ is the total probability of the possible worlds for which, for all j, there are exactly i_j leaves associated with variable x_j.*

The proof is by induction on the tree structure and is omitted.

EXAMPLE 1. *If we associate all leaves with the same variable x, the coefficient of x^i is equal to $\mathsf{Pr}(|pw| = i)$.*

The above can be used to obtain a distribution on the possible world sizes (Figure 1(i)).

EXAMPLE 2. *If we associate a subset S of the leaves with variable x, and other leaves with constant 1, the coefficient of x^i is equal to $\mathsf{Pr}(|pw \cap S| = i)$.*

EXAMPLE 3. *Next we show how to compute $Pr(r(t) = i)$ (i.e., the probability t is ranked at position i), where $r(t)$ denote the* rank *of the tuple in a possible world by some* score *metric. Assume t only has one alternative, (t, a), and hence only one possible value of score, s. Then, in the and/xor tree $\mathcal{T}$, we associate all leaves with key other than t and score value larger than s with variable x, and the leaf (t, a) with variable y, and the rest of leaves with constant 1. Then, the coefficient of $x^{j-1}y$ in the generating function is exactly $Pr(r(t) = i)$. If the tuple has multiple alternatives, we can compute $Pr(r(t) = i)$ for it by summing up the probabilities for the alternatives.*

See Figure 1(iii) for an example.

3.4 Problem Definition

We denote the domain of answers for a query by Ω and the distance function between two answers by $\mathsf{d}()$. Formally, we define the most consensus answer τ to be a feasible answer to the query such that the expected distance between τ and the answer τ_{pw} of the (random) world pw is minimized, i.e,

$$\tau = \arg\min_{\tau' \in \Omega} \{\mathsf{E}[\mathsf{d}(\tau', \tau_{pw})]\}.$$

We call the most consensus answer in Ω *the mean answer* when Ω is the set of all feasible answers. If Ω is restricted to be the set of possible answers (answers of some possible worlds with non-zero probability), we call the most consensus answer in Ω *the median answer*. Taking the example of the top-k queries, the median answer must be the top-k answer of some possible world while the mean answer can be any sorted list of size k.

4. SET DISTANCE MEASURES

We first consider the problem of finding the consensus world for a probabilistic relation under two set distance measures: symmetric difference, and Jaccard distance; the probabilistic relation may be an existing relation in the database, or the result of executing a conjunctive query over it.

4.1 Symmetric Difference

The symmetric difference distance between two sets S_1, S_2 is defined to be

$$\mathsf{d}_\triangle(S_1, S_2) = |S_1 \Delta S_2| = |(S_1 \setminus S_2) \cup (S_2 \setminus S_1)|.$$

Note that two different alternatives of a tuple are treated as different tuples here.

THEOREM 2. *The mean world under the symmetric difference distance is the set of all tuples with probability > 0.5.*

PROOF. Suppose S is a fixed set of tuples and $\bar{S} = T - S$. Let $\delta(p) = \begin{cases} 1, & \text{if } p = true \\ 0, & \text{if } p = false \end{cases}$ be the indicator function. We write $E_{pw \in PW}[\mathsf{d}_\triangle(S, pw)]$ as follows:

$$\begin{aligned} \mathsf{E}[\mathsf{d}_\triangle(S, pw)] &= \mathsf{E}[\sum_{t \in S} \delta(t \notin pw) + \sum_{t \in \bar{S}} \delta(t \in pw)] \\ &= \sum_{t \in S} \mathsf{E}[\delta(t \notin pw)] + \sum_{t \in \bar{S}} \mathsf{E}[\delta(t \in pw)] \\ &= \sum_{t \in S} \mathsf{Pr}(\neg t) + \sum_{t \in \bar{S}} \mathsf{Pr}(t) \end{aligned}$$

Thus, each tuple t contributes $\mathsf{Pr}(\neg t)$ to the expected distance if $t \in S$ and $\mathsf{Pr}(t)$ otherwise, and hence the minimum is achieved by the set of tuples with probability > 0.5. □

Thus, finding the mean answer for a conjunctive query is easy if we can decide which result tuples have probability > 0.5.

Finding the consensus median world is somewhat trickier, with the main concern being that the world that contains all tuples with probability > 0.5 may not be a possible world.

COROLLARY 1. *If the correlations can be modeled using a probabilistic and/xor tree, the median world is the set containing all tuples with probability greater than 0.5.*

The proof is by induction on the height of the tree, and is omitted for space constraints. This however does not hold for arbitrary correlations. Next we show that finding a median answer for a conjunctive query is NP-Hard even if result tuple probability computation is easy (i.e., even if the query has a safe plan) because of the correlations between the result tuples.

THEOREM 3. *For conjunctive queries over databases with arbitrary correlations, finding a median answer is NP-Hard.*

PROOF. Consider the query:

$$Q(C) := \pi_C(R \bowtie S)$$

where $R = R(C, x, b)$ are $S = S(x, b)$ are two relations independent with each other. We show finding a median world for this query is NP-Hard by showing a reduction from the MAX-2-SAT problem. Recall that in a MAX-2-SAT instance, we are given a conjunctive normal form expression with 2 literals per clause and the task is to determine the maximum number of clauses that can be simultaneously satisfied by an assignment. Let the MAX-2-SAT instance consist of n variables, $x_1, \ldots, x_n$, and k clauses. Let $S(x, b) = \{(x_1, 0), (x_1, 1), (x_2, 0), (x_2, 1), \ldots\}$ contain two mutually exclusive tuples each for n variables; all tuples are equiprobable with probability 0.5. $R(C, x, b)$ is a deterministic table, and contains two tuples for each clause: Suppose x_j (or $\bar{x}_j$) is a literal in clause c_i, R contains tuple $(c_i, x_j, 1)$ (or $(c_i, x_j, 0)$). We can see that $R \bowtie S$ has the same set of tuples as R and each tuple has probability 0.5. Moreover, two tuples with the same C value are independent. Therefore, the result of $\pi_C(R \bowtie S)$ contains one tuple for each clause, associated with a probability of $1 - 0.5 \times 0.5 = 0.75$.

Now, consider the possible deterministic answer which is generated by a deterministic instance $\tilde{S}$ of S. It is easy to see the answer contain clause c_i if and only if c_i is satisfied by the assignment defined by $\tilde{S}$. According to the proof of Theorem 2, the median answer is the possible deterministic answer containing maximum number of tuples, which corresponds to finding the assignment that maximizes the number of satisfied clauses. □

4.2 Jaccard Distance

The Jaccard distance between two sets S_1, S_2 is defined to be

$$\mathsf{d}_J(S_1, S_2) = \frac{|S_1 \Delta S_2|}{|S_1 \cup S_2|}.$$

Jaccard distance always lies in $[0, 1]$ and is a real metric, i.e, satisfies triangle inequality. Next we present polynomial time algorithms for finding the mean and median worlds for tuple independent databases, and median world for the BID model.

LEMMA 1. *Given an and/xor tree, $\mathcal{T}$ and a possible world for it, W (corresponding to a set of leaves of $\mathcal{T}$), we can compute $\mathsf{E}[\mathsf{d}(W, pw)]$ in polynomial time.*

PROOF. A generating function $\mathcal{F}_\mathcal{T}$ is constructed with the variables associated with leaves as follows: for $t \in W$ ($t \notin W$), the associated variable is x (y). For example, in a tuple independent database, the generating function is:

$$\mathcal{F}(x, y) = \prod_{t \in W} (\mathsf{Pr}(\neg t) + \mathsf{Pr}(t)x) \prod_{t \notin W} (\mathsf{Pr}(\neg t) + \mathsf{Pr}(t)y)$$

From Theorem 1, the coefficient $c_{i,j}$ of term $x^i y^j$ in generating function $\mathcal{F}$ is equal to the total probability of the worlds such that the Jaccard distance between those worlds and W is exactly $\frac{|W|-i+j}{|W|+j}$. Thus, the distance is $\sum_{i,j} c_{i,j} \frac{|W|-i+j}{|W|+j}$. □

LEMMA 2. *For tuple independent databases, if the mean world contains tuple t_1 but not tuple t_2, then $\mathsf{Pr}(t_1) \geq \mathsf{Pr}(t_2)$.*

PROOF. Say W_1 is the mean world and the lemma is not true, i.e, $\exists t_1 \in W_1, t_2 \notin W_1$ s.t. $\mathsf{Pr}(t_1) < \mathsf{Pr}(t_2)$. Let $W = W_1 - \{t_1\}$, $W_2 = W + \{t_2\}$ and $W' = T - W - \{t_1\} - \{t_2\}$. We will prove W_2 has a smaller expected Jaccard distance, thus rendering contradiction. Suppose $|W_1| = |W_2| = k$. We let matrix $\mathbf{M} = [m_{i,j}]_{i,j}$ where $m_{i,j} = \frac{k-i+j}{k+j}$. We construct generating functions as we did in Lemma 1. Suppose $\mathcal{F}_1$ and $\mathcal{F}_2$ are the generating functions for W_1 and W_2, respectively. We write $||\mathbf{A}|| = \sum_{i,j} a_{i,j}$ for any matrix $\mathbf{A}$ and let $\mathbf{A} \otimes \mathbf{B}$ the Hadamard product of $\mathbf{A}$ and $\mathbf{B}$ (take product entrywise). We denote:

$$\mathcal{F}'(x, y) = \textstyle\prod_{t \in W} (\mathsf{Pr}(\neg t) + \mathsf{Pr}(t)x) \prod_{t \in W'} (\mathsf{Pr}(\neg t) + \mathsf{Pr}(t)y)$$

We can easily see that:

$$\mathcal{F}_1(x, y) = \mathcal{F}'(x, y)\,(\mathsf{Pr}(\neg t_1) + \mathsf{Pr}(t_1)x)\,(\mathsf{Pr}(\neg t_2) + \mathsf{Pr}(t_2)y)$$
$$\mathcal{F}_2(x, y) = \mathcal{F}'(x, y)\,(\mathsf{Pr}(\neg t_1) + \mathsf{Pr}(t_1)y)\,(\mathsf{Pr}(\neg t_2) + \mathsf{Pr}(t_2)x)$$

Then, taking the difference, we get $\bar{\mathcal{F}} = \mathcal{F}_1(x, y) - \mathcal{F}_2(x, y)$ is equal to:

$$\mathcal{F}'(x, y)\,(\mathsf{Pr}(\neg t_1)\mathsf{Pr}(t_2) - \mathsf{Pr}(t_1)\mathsf{Pr}(\neg t_2))\,(y - x) \qquad (1)$$

Let $\mathbf{C}_\mathcal{F} = [c_{i,j}]$ be the coefficient matrix of $\mathcal{F}$ where $c_{i,j}$ is the coefficient of term $x^i y^j$. Using the proof of Lemma 1:

$$\begin{aligned} \mathsf{E}[\mathsf{d}(W_1, pw)] - \mathsf{E}[\mathsf{d}(W_2, pw)] &= ||\mathbf{C}_{\mathcal{F}_1} \otimes \mathbf{M}|| - ||\mathbf{C}_{\mathcal{F}_2} \otimes \mathbf{M}|| \\ &= ||\mathbf{C}_{\bar{\mathcal{F}}} \otimes \mathbf{M}|| \end{aligned}$$

Let $c'_{i,j}$ and $\bar{c}_{i,j}$ be the coefficient of $x^i y_j$ in $\mathcal{F}'$ and $\bar{\mathcal{F}}$, respectively. It is not hard to see $\bar{c}_{i,j} = (c'_{i,j-1} - c'_{i-1,j})p$ from (1) where $p = (\mathsf{Pr}(\neg t_1)\mathsf{Pr}(t_2) - \mathsf{Pr}(t_1)\mathsf{Pr}(\neg t_2)) > 0$.

Then we have:

$$\begin{aligned} ||\mathbf{C}_{\bar{\mathcal{F}}} \otimes \mathbf{M}|| &= p \sum_{i,j} \left((c'_{i,j-1} - c'_{i-1,j}) m_{i,j}\right) \\ &= p \sum_{i,j} c'_{i,j} (m_{i,j+1} - m_{i+1,j}) \\ &= p \sum_{i,j} c'_{i,j} \left(\frac{k-i+j+1}{k+j+1} - \frac{k-i-1+j}{k+j}\right) \end{aligned}$$

The proof follows because, for any $i, j \geq 0$, we have that:

$$\frac{k-i+j+1}{k+j+1} - \frac{k-i-1+j}{k+j} > 0$$ □

The above two lemmas can be used to efficiently find the mean world for tuple-independent databases, by sorting the tuples in the decreasing order by probabilities, and computing the expected distance for every prefix of the sorted order.

A similar algorithm can be used to find the median world for the BID model (by only considering the highest probability alternative for each tuple). Finding mean worlds or median worlds under more general correlation models remains an open problem.

5. TOP-K QUERIES

In this section, we consider top-k queries in probabilistic databases. Each tuple t_i has a score $s(t_i)$. In the tuple-level uncertainty model, $s(t_i)$ is fixed for each t_i, while in the attribute-level uncertainty model, it is an random variable. In the and/xor tree model, we assume that the attribute field is the score (uncertain attributes that don't contribute to the score can be ignored). We further assume no two tuples can take the same score for avoiding ties. We use $r(t)$ to denote the random variable indicating the rank of t and $r_{pw}(t)$ to denote the rank of t in possible world pw. If t does not appear in the possible world pw, then $r_{pw}(t) = \infty$. So,

$\mathsf{Pr}(r(t) > i)$ includes the probability that t's rank is larger than i and that t doesn't exist. We say t_1 *ranks higher* than t_2 in possible world pw if $r_{pw}(t_1) < r_{pw}(t_2)$.

Finally, we use the symbol τ to denote a top-k ranked list, and τ^i to denote the restriction of τ to the first i items. We use $\tau(i)$ to denote the i^{th} item in the list τ for positive integer i, and $\tau(t)$ to denote the position of $t \in T$ in τ.

5.1 Distance between Two Top-k Answers

Fagin et al. [18] provide a comprehensive analysis of the problem of comparing two top-k lists. They present extensions of the Kendall's tau and Spearman footrule metrics (defined on full rankings) to top-k lists and propose several other natural metrics, such as the intersection metric and Goodman and Kruskal's gamma function. In our paper, we consider three of the metrics discussed in that paper: the symmetric difference metric, the intersection metric and one particular extension to Spearman's footrule distance. We briefly recall some definitions here. For more details and the relation between different definitions, please refer to [18].

Given two top-k lists, τ_1 and τ_2, the normalized symmetric difference metric is defined as:

$$\mathsf{d}_\Delta(\tau_1,\tau_2) = \tfrac{1}{2\mathrm{k}}|\tau_1 \Delta \tau_2| = \tfrac{1}{2\mathrm{k}}|(\tau_1 \backslash \tau_2) \cup (\tau_2 \backslash \tau_1)|.$$

While d_Δ focuses only on the membership, the intersection metric d_I also takes the order of tuples into consideration. It is defined to be:

$$\mathsf{d}_I(\tau_1,\tau_2) = \tfrac{1}{\mathrm{k}}\textstyle\sum_{i=1}^{\mathrm{k}} \mathsf{d}_\Delta(\tau_1^i,\tau_2^i)$$

Both $\mathsf{d}_\Delta()$ and $\mathsf{d}_I()$ values are always between 0 and 1.

The original Spearman's Footrule metric is defined as the L_1 distance between two permutations σ_1 and σ_2. Formally, $F(\sigma_1,\sigma_2) = \sum_{t \in T} |\sigma_1(t) - \sigma_2(t)|$. Let ℓ be a integer greater than k. The *footrule distance with location parameter* ℓ, denoted $F^{(\ell)}$ generalizes the original footrule metric. It is obtained by placing all missing elements in each list at position ℓ and then computing the usual footrule distance between them. A natural choice of ℓ is $k+1$ and we denote $F^{(k+1)}$ by d_F. It is also proven that d_F is a real metric and a member of a big and important equivalence class [3] [18].

It is shown in [18] that:

$$\begin{aligned}&\mathsf{d}_F(\tau_1,\tau_2) = (\mathrm{k}+1)|\tau_1 \Delta \tau_2| \\ &+ \sum_{t \in \tau_1 \cap \tau_2} |\tau_1(t) - \tau_2(t)| - \sum_{t \in \tau_1 \backslash \tau_2} \tau_1(t) - \sum_{t \in \tau_2 \backslash \tau_1} \tau_2(t).\end{aligned}$$

Next we consider the problem of evaluating consensus answers for these distance metrics.

5.2 Symmetric Difference and PT-k Ranking Function

In this section, we show how to find mean and median top-k answers under symmetric difference metric in the and/xor tree model. The probabilistic threshold top-k (PT-k) query [26] has been proposed for evaluating ranking queries over probabilistic databases, and essentially returns all tuples t for which $\mathsf{Pr}(r(t) \le \mathrm{k})$ is greater than a given threshold. If we set the threshold carefully so that the PT-k query returns exactly k tuples, we can show that the answer returned is the mean answer under symmetric difference metric.

THEOREM 4. *If* $\tau = \{\tau(1),\tau(2),\ldots,\tau(\mathrm{k})\}$ *is the set of* k *tuples with the largest* $\mathsf{Pr}(r(t) \le \mathrm{k})$*, then* τ *is the mean top-k answer under metric* d_Δ*, i.e., the answer minimizes* $\mathsf{E}[\mathsf{d}_\Delta(\tau,\tau_{pw})]$.

PROOF. Suppose τ is fixed. We write $\mathsf{E}[\mathsf{d}_\Delta(\tau,\tau_{pw})]$ as follows:

$$\begin{aligned}\mathsf{E}[\mathsf{d}_\Delta(\tau,\tau_{pw})] &= \mathsf{E}[\sum_{t \in T} \delta(t \in \tau \wedge t \notin \tau_{pw}) + \delta(t \in \tau_{pw} \wedge t \notin \tau)] \\ &= \sum_{t \in T\backslash\tau} \mathsf{E}[\delta(t \in \tau_{pw})] + \sum_{t \in \tau} \mathsf{E}[\delta(t \notin \tau_{pw})] \\ &= \sum_{t \in T\backslash\tau} \mathsf{Pr}(r(t) \le \mathrm{k}) + \sum_{t \in \tau} \mathsf{Pr}(r(t) > \mathrm{k}) \\ &= \mathrm{k} + \sum_{t \in T} \mathsf{Pr}(r(t) \le \mathrm{k}) - 2\sum_{t \in \tau} \mathsf{Pr}(r(t) \le \mathrm{k})\end{aligned}$$

The first two terms are invariant with respect to τ. Therefore, it is clear that the set of k tuples with the largest $\mathsf{Pr}(r(t) \le \mathrm{k})$ minimizes the expectation. □

To find a median answer, we essentially need to find the top-k answer τ of some possible world such that $\sum_{t \in \tau} \mathsf{Pr}(r(t) \le \mathrm{k})$ is maximum. Next we show how to do this given an and/xor tree in polynomial time.

We write $P(t) = \mathsf{Pr}(r(t) \le \mathrm{k})$ for ease of notation. We can't simply pick k tuples with the highest $P(t)$ values since some of them may be mutually exclusive. We use dynamic programming over the tree structure. For each possible attribute value $a \in A$ (A value is used to rank the tuples in the deterministic setting), let $\mathcal{T}^a$ be the tree which contains all leaves with attribute value at least a. We recursively compute the set of tuples $pw^a(v,i)$, which maximizes the value $\sum_{t \in pw^a(v,i)} P(t)$ among all possible worlds generated by the subtree $\mathcal{T}^a_v$ rooted at v and is of size i, for each node v in $\mathcal{T}^a$ and $1 \le i \le \mathrm{k}$. We compute this for all different a values, and the optimal solution can be chosen to be $\max_a(pw^a(r,\mathrm{k}))$.

Suppose $v_1, v_2, \ldots, v_l$ are v's children. The recursion formula is:

1. If v is a $\textcircled{\vee}$ node,

$$pw^a(v,i) = \arg\max_{pw \in PW(\mathcal{T}^a_{v_i})} \sum_{t \in pw} P(t) = \arg\max_{1 \le j \le l} pw^a(v_j,i).$$

2. If v is a $\textcircled{\wedge}$ node, $pw^a(v,i) = \cup_{1 \le j \le l} pw_j$ such that $pw_j \in PW(\mathcal{T}^a_{v_j})$, $\sum_j |pw_j| = i$ and $\sum_{t \in \cup_j pw_j} P(t)$ is maximized.

In the latter case, the maximum value can be computed by dynamic programming again as follows.

We denote by $pw^a([v_1 \ldots v_h],i)$ the set $\cup_{j=1}^h pw_j$ such that $pw_j \in PW(\mathcal{T}^a_{v_j})$, $\sum_{j=1}^h |pw_j| = i$ and $\sum_{t \in \cup_{j=1}^h pw_j} P(t)$ is maximized. $pw^a([v_1,\ldots v_h],i)$ can also be computed recursively. Let

$$p = \arg\max_{0 \le p \le i} \sum_{t \in pw^a([v_1 \ldots v_{h-1}],p) \cup pw^a(v_h,i-p)} P(t).$$

Then, we have

$$pw^a([v_1 \ldots v_h],i) = pw^a([v_1 \ldots v_{h-1}],p) \cup pw^a(v_h,1-p).$$

Finally, it is easy to see $pw^a(v,i)$ is simply $pw^a([v_1,\ldots,v_l],i)$.

THEOREM 5. *The median top-k answer under symmetric difference metric can be found in polynomial time for a probabilistic and/xor tree.*

5.3 Intersection Metric

Note that the intersection metric d_I is a linear combination of the normalized symmetric difference metric d_Δ. Using a similar

[3] All distance functions in one equivalence class are bounded by each other within a constant factor. This class includes several extensions of Spearman's footrule and Kendall's tau metrics.

approach used in the proof of Theorem 4, we can show that:

$$\mathsf{E}[\mathsf{d}_I(\tau,\tau_{pw})] = \frac{1}{\mathrm{k}}\sum_{i=1}^{\mathrm{k}} \mathsf{E}[\mathsf{d}_\triangle(\tau^i,\tau^i_{pw})]$$

$$= \frac{1}{\mathrm{k}}\sum_{i=1}^{\mathrm{k}} \frac{1}{i}\left(\mathrm{k} + \sum_{t\in T}\mathsf{Pr}(r(t)\le \mathrm{k}) - 2\sum_{t\in\tau^i}\mathsf{Pr}(r(t)\le i)\right)$$

Thus we need to find τ which maximizes the last term, $A(\tau) = \sum_{i=1}^{\mathrm{k}}\left(\frac{1}{i}\sum_{t\in\tau^i}\mathsf{Pr}(r(t)\le i)\right)$. We first rewrite the objective as follows, using the indicator (δ) function:

$$\begin{aligned} A(\tau) &= \sum_{i=1}^{\mathrm{k}}\left(\frac{1}{i}\sum_{t\in T}\mathsf{Pr}(r(t)\le i))\delta(t\in\tau^i)\right) \\ &= \sum_{t\in T}\left(\sum_{i=1}^{\mathrm{k}}\frac{1}{i}\mathsf{Pr}(r(t)\le i)\sum_{j=1}^{i}\delta(t=\tau(j))\right) \\ &= \sum_{t\in T}\sum_{j=1}^{\mathrm{k}}\left(\delta(t=\tau(j))\sum_{i=j}^{\mathrm{k}}\frac{1}{i}\mathsf{Pr}(r(t)\le i)\right) \end{aligned}$$

The last equality holds since $\sum_{i=1}^{\mathrm{k}}\sum_{j=1}^{i} a_{ij} = \sum_{j=1}^{\mathrm{k}}\sum_{i=j}^{k} a_{ij}$.

The optimization task can thus be written as an *assignment problem*, with each tuple t acting as an agent and each of the top-k positions j as a task. Assigning task j to agent t gains a profit of $\sum_{i=j}^{\mathrm{k}}\frac{1}{i}\mathsf{Pr}(r(t)\le i)$ and the goal is to find an assignment such that each task is assigned to at most one agent, and the profit is maximized. The best known algorithm for computing the optimal assignment runs in $O(n\mathrm{k}\sqrt{n})$ time, via computing a maximum weight matching on the bipartite graph [33].

5.4 Approximating the Intersection Metric

We define the following ranking function, where $H_k = \sum_{i=1}^{k} 1/i$ denotes the k^{th} Harmonic number:

$$\Upsilon_H(t) = \sum_{i=1}^{\mathrm{k}}(H_{\mathrm{k}} - H_{i-1})\mathsf{Pr}(r(t)=i) = \sum_{i=1}^{\mathrm{k}}\frac{\mathsf{Pr}(r(t)\le i)}{i}.$$

This is a special case of the parameterized ranking function proposed in [31] and can be computed in $O(n\mathrm{k}\log^2 n)$ time for all tuples in the and/xor tree. We claim that the top-k answer τ_H returned by Υ_H function, i.e., the k tuples with the highest Υ_H values, is a good approximation of the mean answer with respect to the intersection metric by arguing that $\tau_H = \{t_1, t_2, \ldots, t_{\mathrm{k}}\}$ is actually an approximated maximizer of $A(\tau)$. Indeed, we prove the fact that $A(\tau_H) \ge \frac{1}{H_{\mathrm{k}}}A(\tau^*)$ where τ^* is the optimal mean top-k answer.

Let $B(\tau) = \sum_{t\in\tau}\Upsilon_H(t)$ for any top-k answer τ. It is easy to see $A(\tau^*) \le B(\tau^*) \le B(\tau_H)$ since τ_H maximizes the $B()$ function. Then, we can get:

$$A(\tau_H) = \sum_{j=1}^{\mathrm{k}}\sum_{i=j}^{\mathrm{k}}\frac{1}{i}\mathsf{Pr}(r(t_j)\le i)$$

$$\ge \sum_{j=1}^{\mathrm{k}}\left(\frac{H_{\mathrm{k}} - H_{j-1}}{H_k}\right)\sum_{i=1}^{\mathrm{k}}\frac{1}{i}\mathsf{Pr}(r(t_j)\le i)$$

$$= \sum_{j=1}^{\mathrm{k}}\left(\frac{H_{\mathrm{k}} - H_{j-1}}{H_k}\right)\Upsilon_H(t_j) \ge \frac{1}{\mathrm{k}}\sum_{i=1}^{\mathrm{k}}\left(\frac{H_{\mathrm{k}} - H_{i-1}}{H_k}\right)\sum_{i=1}^{\mathrm{k}}\Upsilon_H(t_i)$$

$$= \frac{1}{H_{\mathrm{k}}}B(\tau_H) \ge \frac{1}{H_{\mathrm{k}}}A(\tau^*).$$

The second inequality holds because for non-decreasing sequences $a_i (1\le i\le n)$ and $c_i (1\le i\le n)$,

$$\sum_{i=1}^{n} a_i c_i \ge \tfrac{1}{n}(\sum_{i=1}^{n} a_i)(\sum_{i=1}^{n} c_i)$$

5.5 Spearman's Footrule

For a top-k answer $\tau = \{\tau(1), \tau(2), \ldots, \tau(\mathrm{k})\}$, we define:

- $\Upsilon_1(t) = \sum_{i=1}^{\mathrm{k}}\mathsf{Pr}(r(t)=i)$
- $\Upsilon_2(t) = \sum_{i=1}^{\mathrm{k}}\mathsf{Pr}(r(t)=i)\cdot i$
- $\Upsilon_3(t,i) = \sum_{j=1}^{\mathrm{k}}\mathsf{Pr}(r(t)=j))|i-j| + i\mathsf{Pr}(r(t)>\mathrm{k})$.

It is easy to see $\Upsilon_1(t), \Upsilon_2(t), \Upsilon_3(t)$ can be computed in polynomial time for a probabilistic and/xor tree using our generating functions method.

A careful and non-trivial rewriting of $E_{pw\in PW}[F^*(\tau,\tau_{pw})]$ shows that it also has the form (Figure 2):

$$\mathsf{E}_{pw\in PW}[F^*(\tau,\tau_{pw})] = C + \sum_{t\in T}\sum_{i=1}^{\mathrm{k}}\delta(t=\tau(i))f(t,i)$$

where C is a constant independent of τ, and $f(t,i)$ is a function of t and i that is polynomially computable. More specifically,

$$f(t,i) = \Upsilon_3(t,i) + \Upsilon_2(t) - 2(\mathrm{k}+1)\Upsilon_1(t)$$

Figure 2 shows the exact derivation. Thus, we only need to minimize the second term, which can be modeled as the assignment problem and can be solved in polynomial time.

5.6 Kendall's Tau Distance

Then *Kendall's tau* distance (also called Kemeny distance) d_K between two top-k lists τ_1 and τ_2 is defined to be the number of unordered pairs (t_i, t_j) such that that the order of i and j disagree in any full rankings extended from τ_1 and τ_2, respectively. It is shown that d_F and d_K and a few other generalizations of Spearman's footrule and Kendall's tau metrics form a big equivalence class, i.e., they are within a constant factor of each other [18]. Therefore, the optimal solution for d_F implies constant approximations for all metrics in this class (the constant for d_K is 2).

However, we can also easily obtain a $3/2$-approximation for d_K by extending the $3/2$-approximation for partial rank aggregation problem due to Ailon [1]. The only information used in their algorithm is the proportion of lists where t_i is ranked higher than t_j for all i, j. In our case, this corresponds to $\mathsf{Pr}(r(t_i) < r(t_j))$. This can be easily computed in polynomial time using the generating functions method.

We also note that the problem of optimally computing the mean answer is NP-hard for probabilistic and/xor trees. This follows from the fact that probabilistic and/xor trees can simulate arbitrary possible worlds, and previous work has shown that aggregating even 4 rankings under this distance metric is NP-Hard [16].

6. OTHER TYPES OF QUERIES

We briefly extend the notion of consensus answers to two other types of queries and present some initial results.

6.1 Aggregate Queries

Consider a query of the type:

```
SELECT groupname, count(*)
FROM R
GROUP BY groupname
```

$$\begin{aligned}
\mathsf{E}[F^*(\tau,\tau_{pw})] &= \mathsf{E}\left[(\mathrm{k}+1)|\tau\Delta\tau_{pw}| + \sum_{t\in\tau\cap\tau_{pw}}|\tau(t)-\tau_{pw}(t)| - \sum_{t\in\tau\setminus\tau_{pw}}\tau(t) - \sum_{t\in\tau_{pw}\setminus\tau}\tau_{pw}(t)\right] \\
&= (\mathrm{k}+1)\mathsf{E}[|\tau\Delta\tau_{pw}|] + \sum_{t\in T}\mathsf{E}\left[\delta(t\in\tau\cap\tau_{pw})|\tau(t)-\tau_{pw}(t)|\right] - \sum_{t\in T}\mathsf{E}\left[\delta(t\in\tau\setminus\tau_{pw})\tau(t)\right] - \mathsf{E}\left[\sum_{t\in\tau_{pw}\setminus\tau}\tau_{pw}(t)\right] \\
&= (\mathrm{k}+1)\mathsf{E}[|\tau\Delta\tau_{pw}|] + \sum_{t\in T}\sum_{i=1}^{\mathrm{k}}\sum_{j=1}^{\mathrm{k}}\mathsf{E}\left[\delta(t\in\tau\cap\tau_{pw})\delta(t=\tau_{pw}(i))\delta(t=\tau(j))|i-j|\right] \\
&\quad -\sum_{t\in T}\sum_{i=1}^{\mathrm{k}}\mathsf{E}\left[\delta(t\in\tau\setminus\tau_{pw})\delta(t=\tau(i))i\right] - \sum_{t\in T\setminus\tau}\Upsilon_2(t) \\
&= (\mathrm{k}+1)\mathsf{E}[|\tau\Delta\tau_{pw}|] + \sum_{t\in T}\sum_{i=1}^{\mathrm{k}}\left(\delta(t=\tau(i))\sum_{j=1}^{\mathrm{k}}\mathsf{Pr}(r(t)=j)|i-j|\right) - \sum_{t\in T}\sum_{i=1}^{\mathrm{k}}(\delta(t=\tau(i))i\mathsf{Pr}(r(t)>\mathrm{k})) - \sum_{t\in T\setminus\tau}\Upsilon_2(t) \\
&= (\mathrm{k}+1)(\mathrm{k}+\sum_{t\in T}\Upsilon_1(t) - 2\sum_{t\in\tau}\Upsilon_1(t)) + \sum_{t\in T}\sum_{i=1}^{\mathrm{k}}\delta(t=\tau(i))\Upsilon_3(t,i) - \sum_{t\in T\setminus\tau}\Upsilon_2(t) \\
&= (\mathrm{k}+1)\mathrm{k} + \sum_{t\in T}((\mathrm{k}+1)\Upsilon_1(t)-\Upsilon_2(t)) + \sum_{t\in T}\sum_{i=1}^{\mathrm{k}}\delta(t=\tau(i))(\Upsilon_3(t,i)+\Upsilon_2(t)-2(\mathrm{k}+1)\Upsilon_1(t))
\end{aligned}$$

Figure 2: Derivation for Spearman's Footrule Distance

We assume the dataset is represented by the BID model in which there are m potential groups (indexed by groupname) and n independent tuples with attribute uncertainty. The probabilistic database can be specified by the matrix $\mathrm{P} = [p_{i,j}]_{n\times m}$ where $p_{i,j}$ is the probability that tuple i takes groupname j and $\sum_{j=1}^{m} p_{i,j} = 1$ for any $1 \leq i \leq n$. A query result (on a deterministic relation) is a m-dimensional vector $\mathbf{r}$ where the i^{th} entry is the number of tuples having groupname i. The natural distance metric to use is the squared vector distance.

Computing the mean answer is easy in this case, because of linearity of expectation: we simply take the mean for each aggregate separately, i.e., $\bar{\mathbf{r}} = \mathbf{1}\mathrm{P}$ where $\mathbf{1} = (1, 1, \ldots, 1)$. We note the mean answer minimizes the expected squared vector distance to any possible answer.

The median world requires that the returned answer be a possible answer. It is not clear how to solve this problem optimally in polynomial time. To enumerate all worlds is obviously not computationally feasible. Rounding entries of $\bar{\mathbf{r}}$ to the nearest integers may not result in a possible answer.

Next we present a polynomial time algorithm to find a closest possible answer to the mean world $\bar{\mathbf{r}}$. This yields a 4-approximation for finding the median answer. We can model the problem as follows: Consider the bipartite graph $B(U, V, E)$ where each node in U is a tuple, each node in V is a groupname, and an edge $(u, v), u \in U, v \in V$ indicates that tuple u takes groupname v with non-zero probability. We call a subgraph G' such that $deg_{G'}(u) = 1$ for all $u \in U$ and $deg_{G'}(v) = \mathbf{r}[v]$, an $\mathbf{r}$-*matching* of B for some m-dimensional integral vector $\mathbf{r}$. Given this, our objective is to find an $\mathbf{r}$-matching of B such that $||\mathbf{r} - \bar{\mathbf{r}}||_2^2$ is minimized. Before presenting the main algorithm, we need the following lemma.

LEMMA 3. *The possible world* $\mathbf{r}^*$ *that is closest to* $\bar{\mathbf{r}}$ *is of the following form:* $\mathbf{r}^*[i]$ *is either* $\lfloor\bar{\mathbf{r}}[i]\rfloor$ *or* $\lceil\bar{\mathbf{r}}[i]\rceil$ *for each* $1 \leq i \leq m$.

PROOF. Let M^* be the corresponding $\mathbf{r}^*$-matching. Suppose the lemma is not true, and there exists i such that $|\mathbf{r}^*[i] - \bar{\mathbf{r}}[i]| > 1$. W.l.o.g, we assume $\mathbf{r}^*[i] > \bar{\mathbf{r}}[i]$. The other case can be proved the same way. Consider the connected component $K = \{U', V', E(U', V')\}$ containing i. We claim that there exists $j \in V'$ such that $\mathbf{r}^*[j] < \bar{\mathbf{r}}[j]$ and there is an *alternating path* P with respect to M^* connecting i and j [4]. Therefore, $M' = M^*\Delta P = (M^* \setminus P) \cup (P \setminus M^*)$ is also a valid matching. Suppose M' is a $\mathbf{r}'$-matching. But:

$$\begin{aligned}
||\mathbf{r}' - \bar{\mathbf{r}}||_2^2 &= \sum_{v=1}^{m}(\mathbf{r}'[v] - \bar{\mathbf{r}}[v])^2 \\
&= \sum_{v=1}^{m}(\mathbf{r}^*[v] - \bar{\mathbf{r}}[v])^2 - (\mathbf{r}^*[i] - \bar{\mathbf{r}}[i])^2 - \\
&\quad (\mathbf{r}^*[j] - \bar{\mathbf{r}}[j])^2 + (\mathbf{r}'[i] - \bar{\mathbf{r}}[i])^2 + (\mathbf{r}'[j] - \bar{\mathbf{r}}[j])^2 \\
&= ||\mathbf{r}^* - \bar{\mathbf{r}}||_2^2 - (\mathbf{r}^*[i] - \bar{\mathbf{r}}[i])^2 - (\mathbf{r}^*[j] - \bar{\mathbf{r}}[j])^2 \\
&\quad +(\mathbf{r}^*[i] - 1 - \bar{\mathbf{r}}[i])^2 + (\mathbf{r}^*[j] + 1 - \bar{\mathbf{r}}[j])^2 \\
&= ||\mathbf{r}^* - \bar{\mathbf{r}}||_2^2 + 2 - 2\mathbf{r}^*[i] + 2\bar{\mathbf{r}}[i] + 2\mathbf{r}^*[j] - 2\bar{\mathbf{r}}[j] \\
&< ||\mathbf{r}^* - \bar{\mathbf{r}}||_2^2.
\end{aligned}$$

This contradicts the assumption $\mathbf{r}^*$ is the vector closest to $\bar{\mathbf{r}}$.

Now, we prove the claim. We grow a *alternating path tree* (w.r.t. M^*) rooted at i in a Bread-First-Search (BFS) manner [5]. Let $Odd \subseteq V$ be the set of nodes at odd depth (the root is at depth 1) and $Even \subseteq U$ the set of nodes at even depth. For any subset S of vertices, let $N_B(S)$ denote the set of neighbors of S in graph B. It is easy to see $N_B(Even) = Odd$, $Even \subseteq N_B(Odd)$ and $\sum_{v\in Odd}\mathbf{r}^*[v] = |Even|$. Suppose $\mathbf{r}^*[v] \geq \bar{\mathbf{r}}[v]$ for all v and $\mathbf{r}^*[i] > \bar{\mathbf{r}}[i]$. However, the contradiction follows since:

$$\begin{aligned}
|Even| &= \sum_{v\in Odd}\mathbf{r}^*[v] > \sum_{v\in Odd}\bar{\mathbf{r}}[v] = \sum_{v\in Odd}\sum_{u\in N_B(Odd)}\mathrm{P}[u,v] \\
&= \sum_{v\in Odd}\sum_{u\in Even}\mathrm{P}[u,v] = |Even|.
\end{aligned}$$

[4] An alternating path is a path with alternating unmatched and matched edges [32].

[5] An alternating path tree is a tree in which each path from the root to another node is an alternating path with its first edge being a matched edge[32].

Therefore, there must be a vertex j such that $\mathbf{r}^*[j] < \bar{\mathbf{r}}[j]$ in the alternating path tree. □

With Lemma 3 at hand, we can construct the following min-cost network flow instance to compute the vector $\mathbf{r}^*$ closest to $\bar{\mathbf{r}}$. Add to B a source s and a sink t. Add edges (s, u) with capacity upper bound 1 for all $u \in U$. For each $v \in V$ and $\bar{\mathbf{r}}[v]$ is not integer, add two edges $e_1(v, t)$ and $e_2(v, t)$. $e_1(v, t)$ has both lower and upper bound of capacity $\lfloor \bar{\mathbf{r}}[v] \rfloor$ and $e_2(v, t)$ has capacity upper bound 1 and cost $(\lceil \bar{\mathbf{r}}[v] \rceil - \bar{\mathbf{r}}[v])^2 - (\lfloor \bar{\mathbf{r}}[v] \rfloor - \bar{\mathbf{r}}[v])^2$. If $\bar{\mathbf{r}}[v]$ is a integer, we only add $e_1(v, t)$. We find a min-cost integral flow of value n on this network. For any v such that $e_2(v, t)$ is saturated, we set $\mathbf{r}^*[v]$ to be $\lceil \bar{\mathbf{r}} \rceil$ and $\lfloor \bar{\mathbf{r}} \rfloor$ otherwise. Such a flow with minimum cost suggests the optimality of the vector $\mathbf{r}^*$ due to Lemma 3.

THEOREM 6. *There is a polynomial time algorithm for finding the vector $\mathbf{r}^*$ to $\bar{\mathbf{r}}$ such that $\mathbf{r}^*$ corresponds to some possible answer with non-zero probability.*

Finally, we can prove that:

COROLLARY 2. *There is a polynomial time deterministic 4-approximation for finding the median aggregate answer.*

PROOF. Suppose $\mathbf{r}^*$ is the possible answer closest to the mean answer $\bar{\mathbf{r}}$ and $\mathbf{r}^m$ is the optimal median answer. Let $\mathbf{r}$ be the vector corresponding to the random answer. Then:

$$\begin{aligned}
\mathsf{E}[\mathsf{d}(\mathbf{r}^*, \mathbf{r})] &\leq \mathsf{E}[2(\mathsf{d}(\mathbf{r}^*, \bar{\mathbf{r}}) + \mathsf{d}(\bar{\mathbf{r}}, \mathbf{r}))] \\
&= 2\,(\mathsf{d}(\mathbf{r}^*, \bar{\mathbf{r}}) + \mathsf{E}[\mathsf{d}(\bar{\mathbf{r}}, \mathbf{r})]) \\
&\leq 4\mathsf{E}[\mathsf{d}(\bar{\mathbf{r}}, \mathbf{r})] \leq 4\mathsf{E}[\mathsf{d}(\mathbf{r}^m, \mathbf{r})].
\end{aligned}$$

□

6.2 Clustering

The CONSENSUS-CLUSTERING problem is defined as follows: given k clusterings $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of V, find a clustering $\mathcal{C}$ that minimizes $\sum_{i=1}^{k} \mathsf{d}(\mathcal{C}, \mathcal{C}_i)$. In the setting of probabilistic databases, the given clusterings are the clusterings in the possible worlds, weighted by the existence probability. The main problem with extending the notion of consensus answers to clustering is that the input clusterings are not well-defined (unlike ranking where the score function defines the ranking in any world). We consider a somewhat simplified version of the problem, where we assume that two tuples t_i and t_j are clustered together in a possible world, if and only if they take the same value for the value attribute A (which is uncertain). Thus, a possible world pw uniquely determines a clustering $\mathcal{C}_{pw}$. We define the distance between two clustering $\mathcal{C}_1$ and $\mathcal{C}_2$ to be the number of unordered pairs of tuples that are clustered together in $\mathcal{C}_1$, but separated in the other (the CONSENSUS-CLUSTERING metric). To deal with nonexistent keys in a possible world, we artifically create a cluster containing all of those.

Our task is to find a mean clustering $\mathcal{C}$ such that $\mathsf{E}[\mathsf{d}(\mathcal{C}, \mathcal{C}_{pw})]$. Approximation with factor of $4/3$ is known for CONSENSUS-CLUSTERING [2], and can be adapted to our problem in a straightforward manner. In fact, that approximation algorithm simply needs w_{t_i,t_j} for all t_i, t_j, where w_{t_i,t_j} is the fraction of input clusters that cluster t_i and t_j together, and can be computed as: $w_{t_i,t_j} = \sum_{a \in A} \mathsf{Pr}(i.A = a \wedge j.A = a)$.

To compute these quantities given an and/xor tree, we associate a variable x with all leaves with value (i, a) and (j, a), and constant 1 with the other leaves. From Theorem 1, $\mathsf{Pr}(i.A = a \wedge j.A = a)$ is simply the coefficient of x^2 in the corresponding generating function.

7. CONCLUSION

We addressed the problem of finding a single representative answer to a query over probabilistic databases by generalizing the notion of inconsistent information aggregation. We believe this approach provides a systematic and formal way to reason about the semantics of probabilistic query answers, especially for top-k queries. Our initial work has opened up many interesting avenues for future work. These include design of efficient exact and approximate algorithms for finding consensus answers for other types of queries, exploring connections to safe plans, and understanding the semantics of the other previously proposed ranking functions using this framework.

8. REFERENCES

[1] Nir Ailon. Aggregation of partial rankings, p-ratings and top-m lists. In *SODA*, pages 415–424, 2007.

[2] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. In *J.ACM*, volume 55(5), 2008.

[3] Periklis Andritsos, Ariel Fuxman, and Renee J. Miller. Clean answers over dirty databases. In *ICDE*, 2006.

[4] L. Antova, C. Koch, and D. Olteanu. From complete to incomplete information and back. In *SIGMOD*, 2007.

[5] B., H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE TKDE*, 1992.

[6] G. Beskales, M. Soliman, and I. Ilyas. Efficient search for the top-k probable nearest neighbors in uncertain databases. In *VLDB*, 2008.

[7] J. C. Borda. Mémoire sur les élections au scrutin. *Histoire de l'Acadé Royale des Sciences*, 1781.

[8] R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *SIGMOD*, 2003.

[9] M. J. Condorcet. *Éssai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. 1785.

[10] G. Cormode, F. Li, and K. Yi. Semantics of ranking queries for probabilistic data and expected ranks. In *ICDE*, 2009.

[11] G. Cormode and A. McGregor. Approximation algorithms for clustering uncertain data. In *PODS*, 2008.

[12] Graham Cormode and Minos Garofalakis. Histograms and wavelets on probabilistic data. In *ICDE*, 2009.

[13] N. Dalvi and D. Suciu. Management of probabilistic data: Foundations and challenges. In *PODS*, 2007.

[14] Nilesh Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, 2004.

[15] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, 2004.

[16] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the Tenth International Conference on the World Wide Web (WWW)*, pages 613–622, 2001.

[17] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited. In *Manuscript*, 2001.

[18] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. *SIAM J. Discrete Mathematics*, 17(1):134–160, 2003.

[19] N. Fuhr and T. Rolleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. on Info. Syst.*, 1997.

[20] Gosta Grahne. Horn tables - an efficient tool for handling incomplete information in databases. In *PODS*, 1989.

[21] Todd Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *PODS*, pages 31–40, 2007.

[22] Todd Green and Val Tannen. Models for incomplete and probabilistic information. In *EDBT*, 2006.

[23] Rahul Gupta and Sunita Sarawagi. Creating probabilistic databases from information extraction models. In *VLDB*, Seoul, Korea, 2006.

[24] J. Hodge and R. E. Klima. *The mathematics of voting and elections: a hands-on approach*. AMS, 2000.

[25] M. Hua, J. Pei, W. Zhang, and X. Lin. Efficiently answering probabilistic threshold top-k queries on uncertain data. In *ICDE*, 2008.

[26] M. Hua, J. Pei, W. Zhang, and X. Lin. Ranking queries on uncertain data: A probabilistic threshold approach. In *SIGMOD*, 2008.

[27] T. Imielinski and W. Lipski, Jr. Incomplete information in relational databases. *Journal of the ACM*, 1984.

[28] T. S. Jayram, A. McGregor, S. Muthukrishnan, and E. Vee. Estimating statistical aggregates on probabilistic data streams. In *PODS*, 2007.

[29] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88:571–591, 1959.

[30] L. Lakshmanan, N. Leone, R. Ross, and V. S. Subrahmanian. Probview: a flexible probabilistic database system. *ACM Trans. on DB Syst.*, 1997.

[31] Jian Li, Barna Saha, and Amol Deshpande. A unified approach to ranking in probabilistic databases. http://arxiv.org/abs/0904.1366, 2009.

[32] M.H.Alsuwaiyel. *Algorithms: Design Techniques and Analysis*. World Scienfic, 1998.

[33] S. Micali and V. Vazirani. An $o(sqrt(|v|)|e|)$ algorithm for finding maximum matching in general graphs. In *FOCS*, 1980.

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

[35] C. Ré and D. Suciu. Efficient evaluation of HAVING queries on a probabilistic database. In *DBPL*, 2007.

[36] Christopher Re and Dan Suciu. Materialized views in probabilistic databases for information exchange and query optimization. In *VLDB*, Vienna, Austria, 2007.

[37] A. Sarma, O. Benjelloun, A. Halevy, and J. Widom. Working models for uncertain data. In *ICDE*, 2006.

[38] P. Sen and A. Deshpande. Representing and querying correlated tuples in probabilistic databases. In *ICDE*, 2007.

[39] Prithviraj Sen, Amol Deshpande, and Lise Getoor. Exploiting shared correlations in probabilistic databases. In *VLDB*, 2008.

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

[41] Mohamed A. Soliman and Ihab F. Ilyas. Ranking with uncertain scores. In *ICDE*, 2009.

[42] Y. Wakabayashi. The complexity of computing medians of relations. In *Resenhas*, volume 3(3), pages 323–349, 1998.

[43] D. Wang, E. Michelakis, M. Garofalakis, and J. M. Hellerstein. BayesStore: Managing large, uncertain data repositories with probabilistic graphical models. In *VLDB*, 2008.

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

[45] Ke Yi, Feifei Li, Divesh Srivastava, and George Kollios. Efficient processing of top-k queries in uncertain databases. In *ICDE*, 2008.

[46] Xi Zhang and Jan Chomicki. On the semantics and evaluation of top-k queries in probabilistic databases. In *DBRank*, 2008.