Approximating Predicates and Expressive Queries on Probabilistic Databases

Christoph Koch
Dept. of Computer Science
Cornell University
Ithaca, NY 14853, USA
koch@cs.cornell.edu

ABSTRACT

We study complexity and approximation of queries in an expressive query language for probabilistic databases. The language studied supports the compositional use of confidence computation. It allows for a wide range of new use cases, such as the computation of conditional probabilities and of selections based on predicates that involve marginal and conditional probabilities. These features have important applications in areas such as data cleaning and the processing of sensor data. We establish techniques for efficiently computing approximate query results and for estimating the error incurred by queries. The central difficulty is due to selection predicates based on approximated values, which may lead to the unreliable selection of tuples. A database may contain certain singularities at which approximation of predicates cannot be achieved; however, the paper presents an algorithm that provides efficient approximation otherwise.

Categories and Subject Descriptors

H.2.3 [**DATABASE MANAGEMENT**]: Languages – Query languages; H.2.4 [**DATABASE MANAGEMENT**]: Systems – Query processing

General Terms

Theory, Languages, Algorithms

1. INTRODUCTION

Uncertainty is at the root of many interesting data management problems, in areas such as data extraction, cleaning, and integration, data mining, sensor data management, approximate and online query processing, and scientific data management. Probabilistic databases bear the promise of being useful in all of these areas.

Recently, several groups have established intense programs of work on probabilistic databases. At the University of Washington, the MystiQ project aims at developing scalable query processing techniques, exploiting approximation techniques based on Monte Carlo simulation [7, 16] and following a database theoretic approach of studying structural fragments of query languages that yield efficient evaluation [8]. A group at the University of Maryland is studying the connections between probabilistic databases and previous work in artificial intelligence, specifically Bayesian Networks and graphical models of uncertainty [17]. A group at the University of Florida is currently carring their work on online aggregation in a classical relational model over to probabilistic databases [13]. The Trio project at Stanford has been studying probabilistic databases with a focus on combining uncertainty and data provenance management support [5]. Our own group has had a particular focus on developing more expressive query languages for probabilistic databases that yield new applications [4, 2], while at the same time admitting efficient evaluation [1].

The goal of this paper is to generalize approximation ideas [14, 7, 16] to more expressive, compositional queries in which arbitrary query operations can be executed on top of intermediate result relations that may contain computed, possibly approximated, marginal and conditional probabilities.

The main new problems that arise in such a scenario are twofold. First, while there are established procedures for efficiently approximating the confidence in a tuple, itself a $\#P$-complete problem [7, 10], there is currently no such procedure for deciding predicates over approximated confidence values. And indeed, it is easy to see that in a strict sense such predicates cannot be approximated by a Monte Carlo algorithm. For instance, consider the predicate "confidence $= 1/2$". If an algorithm for approximating the confidence in a tuple seems to converge towards a value much different from $1/2$, we may conclude that with high likelihood the predicate is false. However, we are never able to conclude from the computation of such an algorithm that the probability is exactly $1/2$, even if it is, since the random walk will in general not exactly reach or stabilize on this number. Thus, one major aim of this paper is to obtain an understanding for when predicates can be approximated and to develop algorithms for doing so in those cases.

Second, tuple selection decisions made based on approximated queries may be wrong, leading to a scenario in which tuples are incorrectly present in or absent from an intermediate result. Thus probabilistic databases, which represent uncertainty in the form of weighted sets of possible worlds, can in addition become unreliable. Unreliablility is just another form of uncertainty, and indeed, previous work on query reliability [10, 9] has used a framework essentially

PODS'08, June 9–12, 2008, Vancouver, BC, Canada.

identical to the tuple-independence model present in some recent work on probabilistic databases (e.g., [7]). However, as of yet, there is no framework for understanding probabilistic databases in which approximate query operators may lead to unreliable data. These two problems are the main obstacles on the path towards efficient techniques for approximating the results of more expressive queries, which is the overall goal of this paper.

The structure and contributions of the paper are as follows. Section 2 introduces our model of probabilistic databases and an expressive query algebra. We also provide an example to demonstrate how the constructs of the language enable new, powerful applications of probabilistic databases.

Section 3 provides an actual representation system for probabilistic databases, U-relational databases. This representation system is known to be succinct and complete (i.e., any finite set of possible worlds with probabilities can be represented) and to have the nice property that many operations can be implemented in relational algebra on the representation, by a simple parsimonious translation [1]. We establish complexity bounds for the exact evaluation of general queries with the confidence computation operation. We give a PSPACE-completeness result for combined complexity [19] and $P^{\#P}$-membership and $\#P$-hardness for data complexity (Theorem 3.4). It is interesting that the query language studied in the paper, which is significantly more powerful than the language of [7, 6], is in practice not harder.

Section 4 concisely presents the Karp-Luby Monte Carlo simulation algorithm [14] in its version for approximating tuple confidence.

Section 5 studies the problem of deciding predicates on approximable data values, i.e. values that are the result of approximation and that can be further refined as needed, at a cost. There are four main results. The first establishes a framework for bounding the error probability of a predicate using environments around the data points approximated that are homogeneous with respect to truth of the predicate (Lemma 5.1). The second result gives a complete solution to the problem of maximizing such environments for the case of predicates that are Boolean combinations of linear inequalities (Theorem 5.2), and for establishing error bounds on predicates. The third result gives a complete solution for conditions in which each approximated value is only mentioned once (Theorem 5.5). We then get to discuss singularities at which predicates cannot be approximated and give an algorithm that efficiently approximates predicates everywhere else (Theorem 5.8). The applicability of the results of this section is not restricted to approximate values obtained by the Karp-Luby algorithm or to probabilistic databases, but may conceivably extend to areas such as online aggregation [12, 13].

Section 6 uses the results of the previous section to approximate the results of queries with approximate predicates. We first give a result that provides error bounds on tuples in the results of arbitrary queries (Lemma 6.4). This result uses a notion of data provenance to model the dependence among tuples. We then show that this result yields a polynomially-sized error bound on overall query evaluation that can be reduced to an arbitrarily small error in polynomial time (Proposition 6.6), given that the result contains no tuples that have singularities in their provenance trails. We strengthen this result to admit approximation for individual tuples that do not depend on singularities (Theorem 6.7).

2. PROBABILISTIC DATABASES AND THE ALGEBRA

A schema is a tuple $\langle R_1, \ldots, R_k, c\rangle$, where c is a function mapping each of the relation schemas R_l to either 1 or 0. If $c(R_l) = 1$, then we say that R_l is *by definition* a complete relation. We use superscripts for indexing structures; to avoid confusion with exponentiation, we use bracketed superscripts $.^{[i]}$ for constants.

A *probabilistic database* is a *finite* set of structures

$$\mathbf{W} = \{\langle R_1^1, \ldots, R_k^1, p^{[1]}\rangle, \ldots, \langle R_1^n, \ldots, R_k^n, p^{[n]}\rangle\}$$

of relations R_l^i and numbers $0 < p^{[i]} \leq 1$ such that

$$\sum_{1\leq i\leq n} p^{[i]} = 1$$

and $R_l^1 = \cdots = R_l^n$ if $c(R_l) = 1$. We call an element $\langle R_1^i, \ldots, R_k^i, p^{[i]}\rangle \in \mathbf{W}$ a possible world, and $p^{[i]}$ its probability. It is possible for all possible worlds to agree on further relations than those for which $c(\cdot) = 1$; those marked complete by function c just agree by definition. The *confidence* in tuple $\vec{t}$ is the probability

$$\mathbf{Pr}[\vec{t} \in R] = \sum_{1\leq i\leq n:\ \vec{t}\in R^i} p^{[i]}.$$

Repairing a key of a complete relation R means to compute all subset-maximal relations obtainable from R by removing tuples such that a key constraint is satisfied. We will use this as a method of constructing probabilistic databases, with probabilities derived from relative weights attached to the tuples of R. Formally,

$$\text{repair-key}_{\vec{A}@B}(R) = \{\langle R_f, p_f\rangle \mid f : \pi_{\vec{A}}(R) \to R \text{ s.t. } f(\vec{s}) = \vec{t} \text{ implies } \vec{t}.\vec{A} = \vec{s}\}$$

with

$$R_f = \{\vec{t} \in R \mid f(\vec{t}.\vec{A}) = \vec{t}\}; \quad p_f = \prod_{\vec{t}\in\pi_{\vec{A}}(R)} \frac{f(\vec{t}).B}{\sum_{\vec{s}\in R, \vec{s}.\vec{A}=\vec{t}}\vec{s}.B}.$$

Such a repair operation, apart from its usefulness for the purpose implicit in its name, is a powerful way of constructing probabilistic databases from complete relations. Examples will follow below.

When $\mathbf{W}_1$ and $\mathbf{W}_2$ are two probabilistic databases, we will use $\mathbf{W}_1 \otimes \mathbf{W}_2$ to denote their combination into a single probabilistic database

$$\{\langle \overline{R}, \overline{S}, p\cdot q\rangle \mid \langle \overline{R}, p\rangle \in \mathbf{W}_1, \langle \overline{S}, q\rangle \in \mathbf{W}_2\}. \tag{1}$$

DEFINITION 2.1. Uncertainty algebra (UA) consists of the following operations:

- The operations of relational algebra (selection σ, projection π, product $\times$, union $\cup$, difference $-$, and attribute renaming ρ), which are applied in each possible world independently.

  The semantics of binary operations θ is

  $$[\![\theta(R_l, R_m)]\!](\mathbf{W}) := \{\langle \overline{R}, \theta(R_l, R_m), p\rangle \mid \langle \overline{R}, p\rangle \in \mathbf{W}\}$$

  $$c(\theta(R_l, R_m)) := c(R_l) \wedge c(R_m).$$

  Unary operations can be viewed as binary operations that take twice the same relation as input.

- An operation for computing tuple confidence,

$$[\![\text{conf}(R)]\!](\mathbf{W}) := \{\langle \overline{R}, S, p\rangle \mid \langle \overline{R}, p\rangle \in \mathbf{W}\}$$

with $S = \{\langle \vec{t}, P : \mathbf{Pr}[\vec{t} \in R]\rangle \mid \vec{t} \in \text{poss}(R)\}$, schema $sch(S) = sch(R) \cup \{P\}$ (w.l.o.g., $P \notin sch(R)$), and $c(S) := 1$. Here, $\text{poss}(R) = \bigcup_i R^i$. Note that S is a single relation with a column P for holding probability values, rather than a probabilistic database.

- An uncertainty-introducing operation for repairing a key in a relation. Let $c(R) = 1$ and let column B of R contain only numerical values greater than 0. Then,

$$[\![\text{repair-key}_{\vec{A}@B}(R)]\!](\mathbf{W}) := \mathbf{W} \otimes \text{repair-key}_{\vec{A}@B}(R).$$

We denote fragments of the algebra by $UA[\Theta]$, where Θ denotes the set of operations beyond those of relational algebra that are supported. Positive $UA[\Theta]$ denotes $UA[\Theta]$ without the difference operation. We will also consider an operation $-_c$ which is difference applied to relations that are complete by c. □

We allow for selection conditions that are Boolean combinations of atomic conditions (i.e., negation is permitted even in positive UA) and for arithmetic expressions in atomic conditions and in the arguments of π and ρ. For instance, $\rho_{A+B\to C}(R)$ in each world adds up the A and B values of each tuple of R and keeps them in a new C attribute.

Note that computing possible and certain tuples of a relation is redundant with conf:

$$\begin{aligned}\text{poss}(R) &= \pi_{sch(R)}(\text{conf}(R))\\ \text{cert}(R) &:= \pi_{sch(R)}(\sigma_{P=1}(\text{conf}(R)))\end{aligned}$$

EXAMPLE 2.2 (MOTIVATED BY [11]). We will now use our algebra to compute tables of conditional probabilities. Assume that we have a bag of coins of which we know that it contains two fair coins and one double-headed coin. We take one coin out of the bag but do not look at its two faces to determine its type for certain. Instead we toss the coin twice to collect evidence about its type.

We start out with the following complete database.

Coins	CoinType	Count
	fair	2
	2headed	1

Faces	CoinType	Face	FProb
	fair	H	.5
	fair	T	.5
	2headed	H	1

We pick a coin from the bag and model that the coin be either fair or double-headed.

$$R := \pi_{\text{CoinType}}(\text{repair-key}_{\emptyset@\text{Count}}(\text{Coins}))$$

This results in a probabilistic database consisting of two possible worlds,

R^1	CoinType
	fair

Pr = 2/3

R^2	CoinType
	2headed

Pr = 1/3

In addition, each possible world contains the relations Coins and Faces.

Next we perform the query

$$S := \pi_{\text{CoinType,Toss,Face}}(\text{repair-key}_{\text{CoinType,Toss@FProb}}(\text{Faces} \times \rho_{\text{Toss}}(\{1,2\})))$$

to model the possible outcomes of tossing the chosen coin twice. The probabilistic database representing these repairs consists now of eight possible worlds

$$\langle \text{Coins}, \text{Faces}, R^i, S^j, p^i \cdot (0.5 \cdot 0.5)\rangle,$$

with the four possible relations S_j

S^1	CoinType	Toss	Face
	fair	1	H
	fair	2	H
	2headed	1	H
	2headed	2	H

S^2	CoinType	Toss	Face
	fair	1	T
	fair	2	H
	2headed	1	H
	2headed	2	H

S^3	CoinType	Toss	Face
	fair	1	H
	fair	2	T
	2headed	1	H
	2headed	2	H

S^4	CoinType	Toss	Face
	fair	1	T
	fair	2	T
	2headed	1	H
	2headed	2	H

For instance, the world with relations R^1 and S^1 has probability $2/3 \cdot 1/4 = 1/6$.

The query

$$T := R \bowtie \pi_{\text{CoinType}}(\sigma_{\text{Toss}=1\wedge\text{Face}=H}(S)) \bowtie \pi_{\text{CoinType}}(\sigma_{\text{Toss}=1\wedge\text{Face}=H}(S))$$

computes, for each possible world, the type of the chosen coin if both coin tosses resulted in heads in that world, for the chosen coin type (thus the join with R). Now the query

$$U := \pi_{\text{CoinType},P_1/P_2\to P}(\rho_{P\to P_1}(\text{conf}(T)) \bowtie \rho_{P\to P_2}(\text{conf}(\pi_\emptyset(T))))$$

computes the conditional probability of the chosen coin being of type CoinType given the evidence that we have seen two tosses come out heads up. The resulting relation is

U	CoinType	P
	fair	$\frac{1/6}{1/2} = 1/3$
	2headed	$\frac{1/3}{1/2} = 2/3$

The prior probability of the chosen coin being fair was 2/3; after taking the evidence from two coin tosses into account, the posterior probability $\mathbf{Pr}[\text{the coin is fair} \mid \text{both tosses result in H}]$ is only 1/3. □

$UA[\text{conf}, \text{repair-key}]$ (the maximal language studied in this paper) is a fragment of the query language implemented in the MayBMS system [2, 3]. It also seems to be subsumed by the query language of the Trio System [18]. However, currently no efficient query evaluation techniques are known for this fragment.

3. COMPLEXITY OF QUERIES

To discuss complexity and evaluation of UA, we look at U-relational databases, a representation system for probabilistic databases [1].

A U-relational database defines a weighted set of possible worlds via a finite set of independent discrete random

U_R	CID	LWID	CoinType
	c	fair	fair
	c	2headed	2headed

W	CID	LWID	P
	c	fair	2/3
	c	2headed	1/3

(a) Database after the computation of R.

U_S	CID	LWID	CoinType	Toss	Face
	(fair, 1)	H	fair	1	H
	(fair, 1)	T	fair	1	T
	(fair, 2)	H	fair	2	H
	(fair, 2)	T	fair	2	T
			2headed	1	H
			2headed	2	H

W	CID	LWID	P
	c	fair	2/3
	c	2headed	1/3
	(fair,1)	H	.5
	(fair,1)	T	.5
	(fair,2)	H	.5
	(fair,2)	T	.5

U_T	CID1	LWID1	CID2	LWID2	CID3	LWID3	CoinType
	c	fair	(fair, 1)	H	(fair, 2)	H	fair
	c	2headed					2headed

(b) Database after the computation of T; U_R is as in (a).

Figure 1: U-relational databases.

variables Var. That is, for each $X \in Var$, there is a finite set Dom_X such that, for each $x \in Dom_X$, $\mathbf{Pr}[X = x] > 0$ and $\sum_{x \in Dom_X} \mathbf{Pr}[X = x] = 1$. We use a relation of schema $W(Var, Dom, P)$ with $\langle X, x, p\rangle \in W :\Leftrightarrow \mathbf{Pr}[X = x] = p$ as a complete representation of such a scenario. In addition, a U-relational database consists of a set of representation relations U_R, for each represented relation schema $R(\vec{A})$, of schema $U_R(D, \vec{A})$. Here the D values are partial functions $f : Var \to Dom$, which can be represented as finite sets of pairs of a random variable and a domain value.

A U-relational database $\langle U_{R_1}, \ldots, U_{R_k}, W\rangle$ represents a set of possible worlds not necessarily distinct by the values of their relations but uniquely identifiable by complete functions $f^* : Var \to Dom$ mapping each random variable to a suitable domain value.

A partial function f represents a set of possible worlds with weight

$$p_f = \prod_{X \in Var} \mathbf{Pr}[X = f(X)]. \tag{2}$$

We say that two partial functions f and g are consistent with each other if they agree on those random variables on which they are both defined.

Tuple $\vec{t}$ is in relation R of possible world f^* if there is a tuple $\langle f, \vec{t}\rangle \in U_R$ such that f and f^* are consistent.

It was shown in [1] that any probabilistic database in the sense of the previous section can be represented as a U-relational database.

THEOREM 3.1 ([1]). *U-relational databases are a complete representation system for probabilistic databases.*

EXAMPLE 3.2. Consider again the coin tossing scenario of Example 2.2. Figure 1(a) shows the U-relational database after the computation of relation R. Figure 1(b) shows the database after the computation of S and T, representing eight possible worlds. The final step extends the database by a relation that is complete by definition, thus the U-relation for it is just the relation shown in the previous example. □

U-relational databases have the nice property that the operation of positive relational algebra, poss, and repair-key on probabilistic databases represented as U-relational databases can be evaluated as positive relational algebra queries over the U-relational representations [1]. In summary, the operations translate as follows:

$$\begin{aligned}
[\![R \times S]\!] &:= \pi_{U_R.D \cup U_S.D \to D, sch(R), sch(S)}(\\
&\qquad U_R \bowtie_{U_R.D \text{ cons. with } U_S.D} U_S)\\
[\![\sigma_\phi R]\!] &:= \sigma_\phi(U_R)\\
[\![\pi_{\vec{B}} R]\!] &:= \pi_{D, \vec{B}}(R)\\
[\![R \cup S]\!] &:= U_R \cup U_S\\
[\![\text{poss}(R)]\!] &:= \pi_{sch(R)}(U_R).
\end{aligned}$$

Furthermore, $S := \text{repair-key}_{\vec{A}@B} R$ is translated as

$$U_S := \pi_{D \cup \{(\vec{A}) \mapsto ((sch(R) - \vec{A}) - B)\}, sch(R)} U_R$$

with

$$W := W \cup \pi_{(\vec{A}) \to Var, sch(R) - (\vec{A} \cup \{B\}) \to Dom, B \to P} U_R$$

That is, we introduce new random variables that must be represented in table W. (Note that the remaining operations leave W unchanged.)

We represent the sets D as a fixed set of pairs. If all D values are empty sets of mappings, we use zero columns to represent D, and thus a classical complete relation is a special case of a U-relation. Also, checking consistency can be done by a fixed relational selection in that case, cf. [1].

In [1], it is also shown how attribute-level uncertainty can be realized succinctly by vertical decompositioning without additional cost.

PROPOSITION 3.3 ([1]). *On U-relational databases, the positive UA[repair-key, poss, $-_c$] queries are in LOGSPACE w.r.t. data complexity.*

However, computing the confidence in a tuple of an uncertain relation of a probabilistic database represented as a U-relational database is $\#P$-complete [10, 7].

THEOREM 3.4. *Positive UA[conf, repair-key, $-_c$] on U-relational representations is PSPACE-complete w.r.t. combined complexity and in $P^{\#P}$ and $\#P$-hard w.r.t. data complexity.*

Proof Sketch. The language is obviously PSPACE-hard w.r.t. combined complexity because it contains relational algebra as the special case where the input is a complete

database and we do not make use of the conf and repair-key operations.

For PSPACE-membership, observe that all operations besides conf can be implemented on U-relational representations by just relational algebra. The conf-operation requires a $\#P$-oracle to compute confidence values, but PSPACE is closed under the application of such oracles (which themselves are in PSPACE).

The same query evaluation technique, using relational algebra on U-relational representations on all operations besides conf and using a $\#P$-subprocedure for determining tuple confidence, immediately yields the $P^{\#P}$ bound for data complexity. □

The hardness of confidence computation is due to the succinctness of the representation system. Let us consider the following *nonsuccinct representation* where a probabilistic database is a set of databases with associated weights (as in the definition of probabilistic databases at the beginning of Section 2) and confidence computation is an aggregation operation across this set. Now the algebra – with confidence computation but without repair-key (which in general creates exponentially many new worlds) – has low complexity.

PROPOSITION 3.5. *UA[conf] on nonsuccinct probabilistic databases is in LOGSPACE w.r.t. data complexity.*

4. APPROXIMATING CONFIDENCE

The confidence of tuple $\vec{t}$ for relation R represented in a U-relational database is the weight of $F = \{f \mid \langle f, \vec{t}\rangle \in U_R\}$,

$$p = \sum_{f^*:\ \exists f \in F\ f^* \in \omega(f)} p_{f^*}$$

where $\omega(f)$ denotes the set of complete functions $Var \to Dom$ consistent with partial function f.

We first briefly give a version of the Karp-Luby algorithm [14] for computing the weight p of a disjunction F of partial functions $f : Var \to Dom$.

Let $M = \sum_{f \in F} p_f$ (see Equation 2 for the definition of p_f). Assume an arbitrary fixed order for the elements of F.

DEFINITION 4.1 (KARP-LUBY ESTIMATOR). Consider the following definition of random variable X_i:

1. Choose an f from F with probability p_f/M.
2. Choose a complete function $f^* \in \omega(f)$ with probability p_{f^*}/p_f. That is, on each variable Y on which f is undefined, chose alternative y with probability $\mathbf{Pr}[Y = y]$ according to W.
3. If f is, among the members of F that are consistent with f^*, the one of the smallest index, return 1, otherwise return 0. □

The expected value of X_i is

$$\begin{aligned}\mathbf{E}[X_i] &= \sum_{f \in F} \frac{p_f}{M} \cdot \sum_{f^* \in \omega(f)} \frac{p_{f^*}}{p_f} \cdot \frac{1}{|\{g \mid f^* \in \omega(g)\}|} \\ &= \sum_{f^*:\ \exists f \in F\ f^* \in \omega(f)} \frac{p_{f^*} \cdot |\{f \mid f^* \in \omega(f)\}|}{M \cdot |\{g \mid f^* \in \omega(g)\}|} \\ &= \frac{1}{M} \cdot \sum_{f^*:\ \exists f \in F\ f^* \in \omega(f)} p_{f^*} = \frac{p}{M},\end{aligned}$$

thus X_i is an unbiased estimator for p/M.

The algorithm proceeds by computing the Karp-Luby estimator m times and summing up, $X := \sum_{i=1}^{m} X_i$, with expected value $\mathbf{E}[X] = m \cdot p/M$. We approximate p thus by $\hat{p} := X \cdot M/m$.

Computing X consists of summing up the outcome of m Bernoulli trials. For such a scenario we can use the Chernoff bound

$$\mathbf{Pr}\big[|X - \mathbf{E}[X]| \geq \epsilon \cdot \mathbf{E}[X]\big] \leq 2 \cdot e^{-\epsilon^2 \cdot \mathbf{E}[X]/3}$$

(cf. e.g. [15], Eq. 4.6). By substitution we get

$$\mathbf{Pr}\big[|\hat{p} - p| \geq \epsilon \cdot p\big] = \mathbf{Pr}\Big[\frac{m}{M} \cdot |\hat{p} - p| \geq \epsilon \cdot \frac{m \cdot p}{M}\Big] \leq 2 \cdot e^{-\frac{m \cdot p \cdot \epsilon^2}{3 \cdot M}}$$

and thus, since $p/M \geq 1/|F|$,

$$\delta := \mathbf{Pr}\big[|\hat{p} - p| \geq \epsilon \cdot p\big] \leq 2 \cdot e^{-\frac{m \cdot \epsilon^2}{3 \cdot |F|}}.$$

By choosing

$$m := \Big\lceil \frac{3 \cdot |F| \cdot \log \frac{2}{\delta}}{\epsilon^2} \Big\rceil$$

we get an (ϵ, δ) fully polynomial-time randomized approximation scheme (FPRAS) for confidence computation.

PROPOSITION 4.2 (IMPLICIT IN [14]). *There is a FPRAS for confidence computation.*

Let this new approximate confidence operator be denoted by $\mathrm{conf}_{\epsilon,\delta}$.

COROLLARY 4.3. *Fix ϵ, δ, and query Q of the language of positive UA[$\mathrm{conf}_{\epsilon,\delta}$, repair-key]. Then, Q can be evaluated in polynomial time in the size of the input U-relational database.*

Note that this statement only claims that the evaluation of the operator tree of Q is feasible in polynomial time using randomization. We still have to study the meaning and quality of the query results.

The previous corollary claims efficient evaluation for the positive fragment of UA, using approximation for the difficult confidence operation. One can go a long way with positive queries, but sometimes we would like to compute conditional probabilities of the form $\mathbf{Pr}[\phi \mid \psi] = \mathbf{Pr}[\phi \wedge \psi]/\mathbf{Pr}[\psi]$ where ψ is a universal constraint (e.g. a functional dependency).

The following result shows that many such queries can actually be expressed in the efficiently approximable positive fragment. In the following, it is convenient to use relational calculus terminology. A (slightly generalized) equality-generating dependency (egd) is a Boolean universal formula $\forall \vec{x}\ \phi(\vec{x}) \Rightarrow \psi(\vec{x})$ where ϕ is constructed using atoms, $\wedge$, and $\vee$ and ψ is a Boolean combination of equalities.

THEOREM 4.4. *If π is a formula constructed from existential relational calculus queries and egds using $\wedge$ and $\vee$, then $\mathrm{conf}(\pi)$ is expressible in positive UA[conf].*

Proof Sketch. Consider a conjunction $\phi \wedge \psi$ where ϕ is existential and ψ is an egd.

$$\mathbf{Pr}[\phi \wedge \psi] = \mathbf{Pr}[\phi] - \mathbf{Pr}[\phi \wedge \neg\psi]$$

and $\neg\psi$ is existential. We express this in UA as

$$\rho_{P_1 - P_2 \to P}(\rho_{P \to P_1}(\mathrm{conf}(\phi)) \bowtie \rho_{P \to P_2}(\mathrm{conf}(\phi \wedge \neg\psi))).$$

This idea can be easily generalized to the Boolean combinations claimed in the theorem. □

5. PREDICATES ON APPROXIMABLE VALUES

In this section, we approach the following problem: Given k (possibly different) (ϵ, δ)-approximation schemes for computing approximate values $\hat{p}_1, \ldots, \hat{p}_k$ and a predicate ϕ on those values, how shall we choose ϵ to ensure that the probability of deciding ϕ incorrectly is no greater than δ? An (ϵ, δ) approximation scheme yields approximation to within ϵ times the true value – a relative interval – with probability at least $1 - \delta$, i.e., $\mathbf{Pr}\big[|p - \hat{p}| \geq \epsilon \cdot p\big] \leq \delta$.

This is quite strong, but it does not guarantee, for any fixed ϵ, that the probability of the values created by approximation predicts the predicate with bounded error. We also cannot compute ϵ from ϕ without seeing the data first. Thus we develop an algorithm for approximating the predicate in a small number of iterations in which we look at the data.

Let $\delta_i(\epsilon)$ be an upper bound on the error δ for approximate value $\hat{p}_i$, as a function of ϵ. For instance, for the Karp-Luby algorithm, $\delta_i(\epsilon) = 2 \cdot e^{-\frac{m_i \cdot \epsilon^2}{3 \cdot |F_i|}}$ is such a bound.

LEMMA 5.1. *Let ϕ be a predicate over unreliable attributes modeled as random variables $p_1, \ldots, p_k$. Assume that the values obtained for these are $\hat{p}_1, \ldots, \hat{p}_k$. If $-1 < \epsilon < 1$ is chosen such that the member points of the axis-parallel orthotope defined by the product of open intervals*

$$\Big(\frac{\hat{p}_1}{1+\epsilon}, \frac{\hat{p}_1}{1-\epsilon}\Big) \times \cdots \times \Big(\frac{\hat{p}_k}{1+\epsilon}, \frac{\hat{p}_k}{1-\epsilon}\Big)$$

all agree on $\phi(\cdot)$, then

$$\mathbf{Pr}[\phi(p_1, \ldots, p_k) \neq \phi(\hat{p}_1, \ldots, \hat{p}_k)] \leq \sum_{i=1}^{k} \delta_i(\epsilon).$$

Proof. If $\epsilon > -1$, $\hat{p}_i < (1+\epsilon) \cdot p_i \Leftrightarrow p_i > \hat{p}_i/(1+\epsilon)$. If $\epsilon < 1$, $\hat{p}_i > (1-\epsilon) \cdot p_i \Leftrightarrow p_i < \hat{p}_i/(1-\epsilon)$. Thus, if $-1 < \epsilon < 1$,

$$|p_i - \hat{p}_i| < \epsilon \cdot p_i \Leftrightarrow \frac{\hat{p}_i}{1+\epsilon} < p_i < \frac{\hat{p}_i}{1-\epsilon}.$$

$$\begin{aligned}
&\mathbf{Pr}[\phi(p_1, \ldots, p_k) \neq \phi(\hat{p}_1, \ldots, \hat{p}_k)] = \\
&\qquad \mathbf{Pr}\Big[\neg \bigwedge_{i=1}^{k} \frac{\hat{p}_i}{1+\epsilon} < p_i < \frac{\hat{p}_i}{1-\epsilon}\Big] \\
&= \quad \mathbf{Pr}\Big[\bigvee_{i=1}^{k} |p_i - \hat{p}_i| \geq \epsilon \cdot p_i\Big] \\
&\leq \quad \sum_{i=1}^{k} \mathbf{Pr}\big[|p_i - \hat{p}_i| \geq \epsilon \cdot p_i\big] \quad \leq \quad \sum_{i=1}^{k} \delta_i(\epsilon)
\end{aligned}$$

We can give a slightly better bound if the random variables $p_1, \ldots, p_k$ are independent. Then,

$$\mathbf{Pr}\Big[\bigvee_{i=1}^{k} |p_i - \hat{p}_i| \geq \epsilon \cdot p_i\Big] \leq 1 - \prod_{i=1}^{k} (1 - \delta_i(\epsilon)).$$

The independence assumption is often realistic if the p_i are the results of an approximate computation on a reliable input. For example, the results of multiple applications of the Karp-Luby algorithm are independently distributed. □

If $\delta_i(\epsilon)$ is an exponentially decreasing bound (e.g., a Chernoff bound), then not much is lost by bounding the error by the mass of the area outside such an orthotope.

For application in selection operations, exact attribute values from the database can be viewed as constants for the purpose of the previous lemma.

The goal is now to maximize parameter $\epsilon \geq 0$ in order to minimize the error. The following theorem is a solution for the case that the condition is a linear inequality.

THEOREM 5.2. *Given predicate*

$$\phi(x_1, \ldots, x_k) = \Big(\sum_{i=1}^{k} a_i \cdot x_i \geq b\Big)$$

and a point $(\hat{p}_1, \ldots, \hat{p}_k)$ that satisfies ϕ. Let

$$\alpha = \sum_{i=1}^{k} a_i \cdot \hat{p}_i \qquad \beta = \sum_{i=1}^{k} |a_i \cdot \hat{p}_i|.$$

such that $\alpha \neq 0$. Then,

$$\epsilon = \begin{cases} \alpha/\beta & \ldots \quad b = 0 \\ \max\Big(\frac{\beta}{2b} \pm \frac{1}{2b} \cdot \sqrt{\beta^2 - 4b(\alpha - b)}\Big) & \ldots \quad \textit{otherwise} \end{cases}$$

minimizes the error bound of Lemma 5.1.

Proof. The vector $(a_i)_i$ is orthogonal to the hyperplane

$$h : \sum_i a_i \cdot x_i = b.$$

Let $(y_i)_i$ be the point at which h intersects the line that passes through point $(\hat{p}_i)_i$ and is orthogonal to h. Since $\phi(\hat{p}_1, \ldots, \hat{p}_k)$ is true, either $(a_i)_i = (y_i)_i$ or $(a_i)_i$ points from h towards $(\hat{p}_i)_i$ and $y_i < \hat{p}_i$ iff $a_i > 0$. If we choose ϵ such that the point $(x_i)_i$ with $x_i = \hat{p}_i/(1 + \mathrm{sgn}(a_i \cdot \hat{p}_i) \cdot \epsilon)$ satisfies ϕ, then all points in

$$\Big[\frac{\hat{p}_1}{1+\epsilon}, \frac{\hat{p}_1}{1-\epsilon}\Big] \times \cdots \times \Big[\frac{\hat{p}_k}{1+\epsilon}, \frac{\hat{p}_k}{1-\epsilon}\Big]$$

satisfy ϕ because x_i is the element of the interval $[\hat{p}_i/(1+\epsilon), \hat{p}_i/(1-\epsilon)]$ closest to h. To maximize ϵ, we choose $(x_i)_i$ to be on h. We simplify

$$\sum_i \frac{a_i \cdot \hat{p}_i}{1 + \mathrm{sgn}(a_i \cdot \hat{p}_i) \cdot \epsilon} = b$$

to

$$\sum_i a_i \cdot \hat{p}_i \cdot (1 - \mathrm{sgn}(a_i \cdot \hat{p}_i) \cdot \epsilon) = b \cdot (1-\epsilon) \cdot (1+\epsilon)$$

and further to

$$\alpha - \beta \cdot \epsilon = b - b \cdot \epsilon^2.$$

Thus, if $b = 0$, $\epsilon = \alpha/\beta$. Otherwise, we take the larger of the two solutions of the quadratic equation, i.e.,

$$\epsilon = \max\Big(\frac{\beta}{2b} \pm \frac{1}{2b} \cdot \sqrt{\beta^2 - 4b(\alpha - b)}\Big).$$

Note that since $\beta \geq \alpha \geq b$ and $\alpha \neq 0$, ϵ is always defined, a real number (i.e., the expression under the square root is ≥ 0), and $\epsilon \geq 0$. For $b = 0$, this is obvious. For $b \neq 0$, since $\beta^2 \geq \alpha^2$, $\beta^2 - 4b(\alpha - b) = \beta^2 - \alpha^2 + (\alpha - 2b)^2 \geq 0$. If $b > 0$, then clearly $\epsilon \geq 0$. If $b < 0$, since $\alpha \geq b$, $-4b(\alpha - b) \geq 0$ and $\epsilon \geq 0$. □

REMARK 5.3. Note that if $(\hat{p}_i)_i$ is on the hyperplane h, then the previous result yields $\epsilon = 0$. We will see later

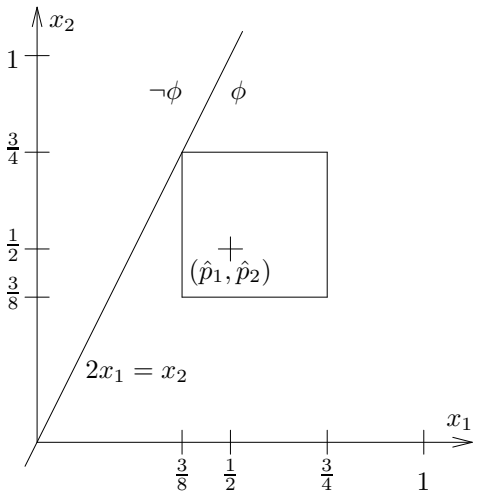

Figure 2: An illustration of Example 5.4. The area to the right of the line $2x_1 = x_2$ contains the points satisfying ϕ. The mass of the area outside the rectangle is used as an upper bound on the error probability.

that this is a case that requires special attention. Values $\epsilon \geq 1$ are also possible in Theorem 5.2 and are inadmissible in the context of the previous lemma and the Karp-Luby algorithm. If such a value is obtained, we chose a value for ϵ which is close to but smaller than 1. □

EXAMPLE 5.4. Suppose that $\phi(x_1, x_2) = (x_1/x_2 \geq c)$ and $\phi(\hat{p}_1, \hat{p}_2) = (\hat{p}_1 - c \cdot \hat{p}_2 \geq 0)$ is true. The error probability is $\mathbf{Pr}[p_1 - c \cdot p_2 < 0] \leq 1 - (1 - B(\epsilon))^2$ where

$$\epsilon = \alpha/\beta = \frac{\hat{p}_1 - c \cdot \hat{p}_2}{\hat{p}_1 + c \cdot \hat{p}_2}.$$

If $\hat{p}_1 = \hat{p}_2 = c = 1/2$, then $\vec{a} = (2, -1)$, $\vec{y} = (.3, .6)$, $\epsilon = 1/3$ and the maximal orthotope is $[3/8; 3/4]^2$, and the point $\vec{x} = (\hat{p}_1/(1+\epsilon), \hat{p}_2/(1-\epsilon))$ at which it touches the hyperplane $2 \cdot x_1 - x_2 = 0$ is $(3/8, 3/4)$. □

In the following, we denote such an ϵ computed for a given predicate ϕ and approximate $(\hat{p}_1, \ldots, \hat{p}_k)$ by $\epsilon_\phi(\hat{p}_1, \ldots, \hat{p}_k)$.

If ϕ is a Boolean combination of inequalities, an ϵ_ϕ can be computed as follows. We first push negations down using De Morgan's law (and the elimination of double negation) and into the inequalities (e.g., $\neg(f(\cdot) < g(\cdot))$ rewrites into $f(\cdot) \geq g(\cdot)$). Then, inductively,

$$\epsilon_{\phi \wedge \psi}(\hat{p}_1, \ldots, \hat{p}_k) := \min(\epsilon_\phi(\hat{p}_1, \ldots, \hat{p}_k), \epsilon_\psi(\hat{p}_1, \ldots, \hat{p}_k))$$
$$\epsilon_{\phi \vee \psi}(\hat{p}_1, \ldots, \hat{p}_k) := \max(\epsilon_\phi(\hat{p}_1, \ldots, \hat{p}_k), \epsilon_\psi(\hat{p}_1, \ldots, \hat{p}_k))$$

We next develop a result that gives rise to an algorithm for maximizing ϵ in predicates defined by general algebraic inequalities, with the only restriction that each variable must only occur once. While this may seem like a serious constraint on expressiveness, it is really means only a small loss of efficiency: rather than using the same unreliable value twice in a formula, we can instead approximate the same value twice (yielding a value with an independenty error) and represent the two approximation results by two different variables. Thus, the following theorem yields a general solution for algebraic inequalities.

THEOREM 5.5. *Given a constant $\epsilon > 0$ and a predicate*

$$\phi(x_1, \ldots, x_k) = (f(x_1, \ldots, x_k) \geq 0)$$

where f is an algebraic expression built from constants, exactly one occurrence of each of the variables $x_1, \ldots, x_k$, and the operations $+, -, \cdot$, and $/$. Then, if each of the corner points of the orthotope

$$\left[\frac{\hat{p}_1}{1+\epsilon}, \frac{\hat{p}_1}{1-\epsilon}\right] \times \cdots \times \left[\frac{\hat{p}_k}{1+\epsilon}, \frac{\hat{p}_k}{1-\epsilon}\right]$$

agrees with point $(\hat{p}_1, \ldots, \hat{p}_k)$ on ϕ, then so do all points in the orthotope.

Proof. We prove the following stronger result.

Claim: Whenever

$$\phi(x_1, \ldots, x_k) = \phi(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_k)$$

(w.l.o.g., $x_i \leq x_i'$) implies

$$\phi(x_1, \ldots, x_k) = \phi(x_1, \ldots, x_{i-1}, x_i'', x_{i+1}, \ldots, x_k)$$

for all points in the orthotope such that $x_i \leq x_i'' \leq x_i'$, then all the points in the orthotope agree on ϕ.

Indeed, let $(z_1, \ldots, z_k)$ be an arbitrary point in the orthotope. For $1 \leq i \leq k$, if

$$\phi(z_1, \ldots, z_{i-1}, \frac{\hat{p}_i}{1+\epsilon}, \frac{\hat{p}_{i+1}}{1+\chi_{i+1}\cdot\epsilon}, \ldots, \frac{\hat{p}_k}{1+\chi_k\cdot\epsilon}) =$$

$$\phi(z_1, \ldots, z_{i-1}, \frac{\hat{p}_i}{1-\epsilon}, \frac{\hat{p}_{i+1}}{1+\chi_{i+1}\cdot\epsilon}, \ldots, \frac{\hat{p}_k}{1+\chi_k\cdot\epsilon})$$

for all $\chi_{i+1}, \ldots, \chi_k \in \{+1, -1\}$, then

$$\phi(z_1, \ldots, z_{i-1}, \frac{\hat{p}_i}{1+\epsilon}, \frac{\hat{p}_{i+1}}{1+\chi_{i+1}\cdot\epsilon}, \ldots, \frac{\hat{p}_k}{1+\chi_k\cdot\epsilon}) =$$

$$\phi(z_1, \ldots, z_{i-1}, z_i, \frac{\hat{p}_{i+1}}{1+\chi_{i+1}\cdot\epsilon}, \ldots, \frac{\hat{p}_k}{1+\chi_k\cdot\epsilon})$$

and thus $(z_1, \ldots, z_k)$ agrees on ϕ with the corner points. An easy induction proves the claim. As $(z_1, \ldots, z_k)$ was an arbitrary point from the orthotope, it also agrees on ϕ with $(\hat{p}_1, \ldots, \hat{p}_k)$.

Now consider Boolean functions

$$f : x_i \mapsto \phi(a_1, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_k)$$

where the $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k$ are constants. Obviously, if all such functions definable over predicate ϕ are monotonic, then the precondition of the previous claim holds.

But this is obviously the case for the predicates ϕ of our theorem: If we fix all variables but one to constants, since each variable occurs only once in ϕ, the result will be of one of two forms, $a \cdot x_i + b \geq 0$, or $a/x_i + b \geq 0$, where a and b are constants. Obviously, both forms of Boolean functions are monotonic. □

Thus, ϵ can be maximized by binary search in the interval $(0, 1)$, checking in each step whether the candidate ϵ satisfies the requirement that all the 2^k (where k is fixed with the predicate) corner points of the orthotope agree with $(\hat{p}_1, \ldots, \hat{p}_k)$ on ϕ.

There is, however, a fundamental problem with bounding the error of predicates on unreliable attributes. In some cases, no matter how small an $\epsilon > 0$ we choose, we cannot separate the point $(\hat{p}_1, \ldots, \hat{p}_k)$ enough from a boundary (a

foreach i **do** { $X_i := 0$; $m_i := 0$; }
do {
  foreach i **do** {
    repeat $|F_i|$ times **do**
      $X_i := X_i$ + Karp-Luby-estimator(F_i);
    $m_i := m_i + |F_i|$; $\hat{p}_i := X_i \cdot M_i/m_i$;
  }
  if $\phi(\hat{p}_1, \ldots, \hat{p}_k)$ is true **then**
    $\epsilon := \max(\epsilon_0, \epsilon_\phi(\hat{p}_1, \ldots, \hat{p}_k))$;
  else
    $\epsilon := \max(\epsilon_0, \epsilon_{\neg\phi}(\hat{p}_1, \ldots, \hat{p}_k))$;
}
until $\sum_i \delta_i(\epsilon) \leq \delta$;

output $\phi(\hat{p}_1, \ldots, \hat{p}_k)$, error bound $\min(0.5, \sum_i \delta_i(\epsilon))$

Figure 3: Predicate approximation algorithm.

hyperplane in the case of atomic conditions that are linear inequalities, cf. Theorem 5.2) at which the truth value of the predicate changes; this is the case if $(p_1, \ldots, p_k)$ lies exactly on such a boundary. We will call such boundary points *singularities*.

DEFINITION 5.6 (ϵ_0-SINGULARITY). A point $(p_1, \ldots, p_k)$ is called an ϵ_0-singularity if there is a point $(x_1, \ldots, x_k)$ such that $\bigwedge_i |p_i - x_i| \leq \epsilon_0 \cdot p_i$ and $\phi(p_1, \ldots, p_k) \neq \phi(x_1, \ldots, x_k)$.

EXAMPLE 5.7. Consider predicate $x_1 \geq c$. If $p_1 = c$, we have a singularity. As a consequence, we can in particular never approximate a tuple certainty test ($c = 1$), no matter how we set up ϵ. We are able to detect cases where $p_1 < 1$, but we will never be able to tell for sure that $p_1 = 1$. □

The results of this section yield a method of deciding a predicate with probability at least δ except in the case of a singularity. We will look in detail at the case in which all our approximable values are tuple confidences and we use the Karp-Luby algorithm with error bound $\delta_i(\epsilon) = 2 \cdot e^{-\frac{m_i \cdot \epsilon^2}{3 \cdot |F_i|}}$ for approximation. Let $\epsilon_0 > 0$ be the smallest that we are willing the approximation technique to go for. A naive procedure is to compute each $\hat{p}_i$ using $m = 3|F| \cdot \log(2/\delta)/\epsilon^2$. Let $\psi = \phi$ if $\phi(\hat{p}_1, \ldots, \hat{p}_k)$ is true and $\neg\phi$ otherwise. If $\epsilon_\psi(\hat{p}_1, \ldots, \hat{p}_k) \geq \epsilon_0$, then $\phi(\hat{p}_1, \ldots, \hat{p}_k) = \phi(p_1, \ldots, p_k)$, i.e., our answer for ϕ is correct, with probability at least $1 - \delta$. This does not exploit the fact that if $\epsilon_\psi(p_1, \ldots, p_k) > \epsilon_0$, we can decide ϕ with sufficiently low error even earlier. The algorithm shown in Figure 3 does.[1]

The next theorem asserts that this algorithm indeed approximates the predicate unless it essentially cannot be approximated because the true value that we are approximating constitutes a singularity.

THEOREM 5.8. *On input of $F_1, \ldots, F_k$, ϵ_0, and δ, if point $(p_1, \ldots, p_k)$ is not an ϵ_0-singularity, then the algorithm of Figure 3 computes $\phi(p_1, \ldots, p_k)$ with error probability $\leq \delta$.*

Proof Sketch. Suppose that for all points $(x_1, \ldots, x_k)$ with $|p_i - x_i| \leq \epsilon_0 \cdot p_i$ for all i, $\phi(p_1, \ldots, p_k) \Leftrightarrow \phi(x_1, \ldots, x_k)$. Let $\psi = \phi$ if $\phi(\hat{p}_1, \ldots, \hat{p}_k)$ is true and $\psi = \neg\phi$ otherwise.

[1]The Karp-Luby estimator was given in Definition 4.1. Note that there i was a different index from the one used in the algorithm of Figure 3.

There are two cases. (1) The algorithm terminates early, with $\epsilon = \epsilon_\psi(\hat{p}_1, \ldots, \hat{p}_k) \geq \epsilon_0$. Then the probability that at least one of the p_i is outside the range $[\hat{p}_i/(1+\epsilon), \hat{p}_i/(1-\epsilon)]$ is no greater than δ. Thus, for the algorithm to make a wrong decision, $|p_i - \hat{p}_i| > (\epsilon + \epsilon_0) \cdot p_i$, which is even less likely. (2) The algorithm terminates and $\epsilon_\psi(\hat{p}_1, \ldots, \hat{p}_k) < \epsilon_0$. While we did not succeed in getting sufficient support for deciding our predicate, we have nevertheless run the Karp-Luby algorithm and it is assured that the probability that $|p_i - \hat{p}_i| \geq \epsilon_0 \cdot p_i$ for any i is true is no greater than δ. But then, by our assumption, $\phi(p_1, \ldots, p_k) \Leftrightarrow \phi(\hat{p}_1, \ldots, \hat{p}_k)$. In both cases the error probability is bounded by δ, thus it is so overall. □

The algorithm keeps the individual errors $\delta_i(\epsilon)$ balanced, with $\delta_1(\epsilon) = \cdots = \delta_k(\epsilon) = 2 \cdot e^{-l \cdot \epsilon^2/3}$, where l is the number of iterations of the outer loop. Let us denote

$$\delta'(\epsilon, l) := 2 \cdot e^{-l \cdot \epsilon^2/3}.$$

The overall number of invocations of the Karp-Luby estimator is $l \cdot \sum_i |F_i|$, with $l = \lceil 3 \log(2k/\delta)/\epsilon^2 \rceil$ and $\epsilon \geq \epsilon_0$. The running time improves by close to[2] a factor of $(\epsilon_\phi^2 - \epsilon_0^2)/\epsilon_\phi^2$ over the naive algorithm sketched above.

6. APPROXIMATING QUERIES

Now that we have obtained a method for approximating predicates in those cases where they can be approximated, we look for an approximation algorithm for UA queries.

Basically, all the building blocks for the approximate evaluation of UA queries are available. We can use U-relational databases as the representation system, which yields efficient techniques for evaluating the operations of positive relational algebra. For confidence computation, we have the Karp-Luby algorithm, and for approximating a selection predicate we have the algorithm of Figure 3. The only piece that is missing is to know the parameters for the approximation operators that will guarantee that the overall error bound does not exceed a given δ. That is, we are looking for the parameters m respectively l of iterations that will be required in each of the applications of an approximation operator. This yields two questions: Can bounds be given for these parameters, and are they polynomial, yielding polynomial-time query evaluation overall?

In order to approach these questions, we simplify the query language. The goal is to distill a language that allows for proofs that provide some insight; the language captures the interaction of selection based on approximate values with relational algebra operations and will disregard repair-key operations and the construction of approximate confidence values for the output.

We introduce a new *approximate selection* operation

$$\hat{\sigma}_{\phi(\mathrm{conf}_{\epsilon,\delta}[\vec{A}_1], \ldots, \mathrm{conf}_{\epsilon,\delta}[\vec{A}_k])}(R) := \sigma_{\phi(P_1,\ldots,P_k)}(\rho_{P \to P_1}(\mathrm{conf}_{\epsilon,\delta}(\pi_{\vec{A}_1}(R))) \bowtie \ldots \bowtie \rho_{P \to P_k}(\mathrm{conf}_{\epsilon,\delta}(\pi_{\vec{A}_k}(R)))).$$

EXAMPLE 6.1. The query

$$\sigma_{P_1/P_2 \leq .5}(\rho_{P \to P_1}(\mathrm{conf}_{\epsilon,\delta}(T)) \bowtie \rho_{P \to P_2}(\mathrm{conf}_{\epsilon,\delta}(\pi_\emptyset(T))))$$

is now written as $\hat{\sigma}_{\mathrm{conf}[CT]/\mathrm{conf}[\emptyset] \leq 0.5}(T)$. □

[2]Due to the fact that the truth value $\phi(\hat{p}_1, \ldots, \hat{p}_k)$ may switch as $(\hat{p}_1, \ldots, \hat{p}_k)$ becomes more and more exact.

We will study positive $UA[\hat{\sigma}]$ throughout this section.

In addition to complete relations (cf. function c), we keep track of *unreliable* relations. A relation is unreliable if it was created using approximate selection or by an arbitrary operation whose input was unreliable.

We will use Q to denote a query that, however, uses exact implementations of the confidence operation in $\hat{\sigma}$; we will use $Q^{\sim}$ to denote the same query that uses the approximate operator implementations as just defined, using $\mathrm{conf}_{\epsilon,\delta}$.

To study the probability of error in positive $UA[\hat{\sigma}]$ queries, we require a model of unreliability. Unreliability is a form of uncertainty, and we will model this by restating the approximate selection operation as an uncertainty-introducing operation.

DEFINITION 6.2. An uncertain, unreliable database is a probabilistic database of the form $\mathbf{F} \otimes \mathbf{G}$ (see Eq. 1 of Section 2), where $\mathbf{F}$ is called its uncertain and $\mathbf{G}$ its unreliable component.

Approximate selection is defined by an unreliability-to-uncertainty transformation. Starting from a (possibly unreliable) complete relation R, we construct an uncertain relation with the tuple independence model. Each tuple $\vec{t}$ of R is independently in the result with probability $\geq 1-\delta$ if it is in the result of the approximate selection, and not in the result with probability $\geq 1-\delta$ if it is not in the result of the approximate selection. (We have thus completely specified all four cases.) □

EXAMPLE 6.3. Consider a relation R that contains two tuples t_1, t_2 with confidence values approximated using error bound δ. Based on the approximate values, an approximate selection operation $\hat{\sigma}_\phi$ selects t_2 and drops t_1. One might assume that this can be modeled by an uncertain relation R' that contains t_1 with probability $\leq \delta$ and t_2 with probability $\geq 1-\delta$: δ is only an upper bound on the error probability, not the error probability itself. Assume that the true error probability is $0 < e < \delta$ for t_1 and δ for t_2. Then $\mathbf{Pr}[\sigma_\phi(R) \neq \emptyset] = 1-\delta+e\delta$ while $\mathrm{conf}(\pi_\emptyset(R')) = 1-\delta+\delta^2$, which is too great and will lead to a too small error bound. This shows that we cannot just view unreliability with *bounds on error probability* as the same scenario as unreliability with error probabilities as studied in [10]. □

We define *provenance* as a relationship $\prec$ between (tuple, query)-pairs obtained by the transitive closure of the relation

$$\begin{aligned}
(t.\vec{A}, \pi_{\vec{A}}(R)) &\prec (t, R)\\
(t, \sigma_\phi(R)) &\prec (t, R)\\
(t, R \cup S) &\prec (t, R)\\
(t, R \cup S) &\prec (t, S)\\
(\langle r, s\rangle, R \times S) &\prec (r, R)\\
(\langle r, s\rangle, R \times S) &\prec (s, S)
\end{aligned}$$

Intuitively, $(t, Q) \prec (r, R)$ is true if there exists a database in which changing the membership of r in R changes the membership of t in the result of the positive relational algebra query Q on the database.

The next lemma provides bounds on the error of arbitrary positive $UA[\hat{\sigma}]$ expressions. Because of space limitations, we refer to its proof sketch for the inlined definitions of $\mathbf{Pr}_{\mathbf{F}\otimes\mathbf{G}}$ and $\mathbf{Pr}_{\mathbf{G}}$.

LEMMA 6.4. *Let $Q^{\sim}$ be a query in positive $UA[\hat{\sigma}]$ over an uncertain, unreliable database $\mathbf{F} \otimes \mathbf{G}$.*

1. $\mathbf{Pr}_{\mathbf{F}\otimes\mathbf{G}}[\vec{t} \in Q \not\Leftrightarrow \vec{t} \in Q^{\sim}] \leq \sum_{(\vec{t},Q)\prec(\vec{s},\hat{\sigma}_\phi(Q_0))} \mathbf{Pr}_{\mathbf{F}\otimes\mathbf{G}}[\vec{s} \in Q_0 \not\Leftrightarrow \vec{s} \in Q_0^{\sim}]$. *Here, the queries $\hat{\sigma}_\phi(Q_0)$ are the maximal $\hat{\sigma}$-subexpressions of Q.*

2. $\mathbf{Pr}_{\mathbf{G}}[\vec{t} \in \hat{\sigma}_{\phi(f_1,\ldots,f_k)}(Q) \not\Leftrightarrow \vec{t} \in \hat{\sigma}_{\phi(f_1,\ldots,f_k)}(Q^{\sim})] \leq k \cdot \delta'(\max(\epsilon_\phi, \epsilon_0), l) + \mathbf{Pr}_{\mathbf{F}\otimes\mathbf{G}}[\vec{t} \in Q \not\Leftrightarrow \vec{t} \in Q^{\sim}]$.

Proof (Rough Sketch). We will make use of the fact that an input uncertain, unreliable database $\mathbf{F} \otimes \mathbf{G}$ has its uncertain and unreliable component independent from each other. The operations of relational algebra can produce tuples that depend on both uncertain and unreliable data, but the conf and approximate selection operations close the possible worlds semantics on the side of $\mathbf{F}$ and their output relations are again complete but unreliable. Unreliable data cannot flow to the uncertain side, and we can, at the pivotal unreliable operations, always again produce a factored database $\mathbf{F} \otimes \mathbf{G}$.[3]

(1) We use $\chi[\cdot]$ to map true conditions to 1 and false conditions to 0. We use $f \in \mathbf{F}$, $g, g_0 \in \mathbf{G}$ to identify possible worlds; g_0 denotes the correct world in the set $\mathbf{G}$ representing the unreliable part of the database. $\hat{\sigma}_\phi(Q_0)$ denotes maximal (since $\prec$ is not defined for $\hat{\sigma}$) $\hat{\sigma}$-subexpressions of Q. $Q^{f,g}$ denotes the result of query Q in world (f, g).

$$\begin{aligned}
&\mathbf{Pr}_{\mathbf{F}\otimes\mathbf{G}}[\vec{t} \in Q \not\Leftrightarrow \vec{t} \in Q^{\sim}] :=\\
&\qquad \sum_{f\in\mathbf{F}, g\in\mathbf{G}} p_f \cdot p_g \cdot \chi[\vec{t} \in Q^{f,g_0} \not\Leftrightarrow \vec{t} \in Q^{f,g}]\\
&\leq \sum_{f\in\mathbf{F}, g\in\mathbf{G}} p_f \cdot p_g \cdot\\
&\qquad \chi\Big[\bigvee_{(\vec{t},Q)\prec(\vec{s},\hat{\sigma}_\phi(Q_0)), \vec{s}\in poss(\hat{\sigma}_\phi(Q_0))} \vec{s} \in Q_0^{f,g_0} \not\Leftrightarrow \vec{s} \in Q_0^{f,g}\Big]\\
&\leq \sum_{(\vec{t},Q)\prec(\vec{s},\hat{\sigma}_\phi(Q_0)), \vec{s}\in poss(\hat{\sigma}_\phi(Q_0))} \mathbf{Pr}_{\mathbf{F}\otimes\mathbf{G}}[\vec{s} \in Q_0 \not\Leftrightarrow \vec{s} \in Q_0^{\sim}]
\end{aligned}$$

For the definition of p_f, p_g see Equation 2.

(2) $\mathbf{Pr}_{\mathbf{G}}[\mathrm{conf}(\vec{t} \in Q) \neq \mathrm{conf}(\vec{t} \in Q^{\sim})] :=$

$$\begin{aligned}
&\sum_{g\in\mathbf{G}} p_g \cdot \chi\Big[\sum_{f\in\mathbf{F}} p_f \cdot \chi[\vec{t} \in Q^{f,g_0}] \neq \sum_{f\in\mathbf{F}} p_f \cdot \chi[\vec{t} \in Q^{f,g}]\Big]\\
&\leq \sum_{f\in\mathbf{F}, g\in\mathbf{G}} p_f \cdot p_g \cdot \chi[\vec{t} \in Q^{f,g_0} \not\Leftrightarrow \vec{t} \in Q^{f,g}]\\
&= \mathbf{Pr}_{\mathbf{F}\otimes\mathbf{G}}[\vec{t} \in Q \not\Leftrightarrow \vec{t} \in Q^{\sim}]
\end{aligned}$$

By the algorithm of Figure 3,

$$\begin{aligned}
&\mathbf{Pr}[\phi(f_1(\vec{t}, Q^{\sim}), \ldots, f_k(\vec{t}, Q^{\sim})) \not\Leftrightarrow\\
&\quad \phi(f_1^{\sim}(\vec{t}, Q^{\sim}), \ldots, f_k^{\sim}(\vec{t}, Q^{\sim}))] \leq k \cdot \delta'(\max(\epsilon_\phi, \epsilon_0), l).
\end{aligned}$$

Therefore,

[3]This would not be so if repair-key could use unreliable relations. However, the results of this paper are immediately applicable to queries that also use repair-key operations, however never above an approximate selection operation in the algebra tree.

$$\begin{aligned}
&\mathbf{Pr}_{\mathbf{G}}[\vec{t} \in \hat{\sigma}_{\phi(f_1,\ldots,f_k)}(Q) \not\Leftrightarrow \vec{t} \in \hat{\sigma}_{\phi(f_1,\ldots,f_k)}(Q^{\sim})] \;\leq \\
&\qquad \mathbf{Pr}[\phi(f_1(\vec{t},Q^{\sim}),\ldots,f_k(\vec{t},Q^{\sim})) \not\Leftrightarrow \\
&\qquad\quad \phi(f_1^{\sim}(\vec{t},Q^{\sim}),\ldots,f_k^{\sim}(\vec{t},Q^{\sim}))] \\
&\quad + \quad \mathbf{Pr}_{\mathbf{G}}[\mathrm{conf}(\vec{t},Q) \neq \mathrm{conf}(\vec{t},Q^{\sim})] \\
&\quad \leq \quad k \cdot \delta'(\max(\epsilon_\phi,\epsilon_0),l) + \mathbf{Pr}_{\mathbf{F}\otimes\mathbf{G}}[\vec{t} \in Q \not\Leftrightarrow \vec{t} \in Q^{\sim}]. \quad \square
\end{aligned}$$

EXAMPLE 6.5. We discuss an extreme example in which the provenance of a result tuple consists of the entire input. Consider an unreliable relation $R(AB)$ that is empty but could independently contain each one of the tuples $\langle a, b_i\rangle$, $1 \leq i \leq n$, each at probability μ. Let g_0 be the one possible world in which the query $\pi_A(R)$ does not return the tuple $\langle a\rangle$. Thus, the probability $\mathbf{Pr}[\langle a\rangle \in \pi_A(R) \not\Leftrightarrow \langle a\rangle \in \pi_A(R^{\sim})]$ is $1 - p_{g_0} = 1 - (1-\mu)^n \leq \mu \cdot n$. $\square$

Now, if we define provenance in addition also for approximate selections,

$$(\vec{t}, \hat{\sigma}_{\phi,f_1,\ldots,f_k}(Q)) \prec (\vec{t}, Q),$$

then we can link result tuples to tuples in their provenance that cause singularities in approximate selections.

PROPOSITION 6.6. *Given a positive $UA[\hat{\sigma}]$ query Q of nesting depth d of approximate selection operators and an integer k that is an upper bound both on the maximum arity of the relations defined by subqueries and the number of unreliable attributes used in any single approximate selection operator, and n is the number of active domain elements in the database, the probability $\mathbf{Pr}[\vec{t} \in Q \not\Leftrightarrow \vec{t} \in Q^{\sim}]$ can be bounded by $k \cdot d \cdot n^{k \cdot d} \cdot \delta'(\epsilon_0, l)$ if $\vec{t}$ does not have singularities in its provenance.*

Proof Sketch. Rephrasing Lemma 6.4, the error probability $\mu(R)$ of tuples in a reliable relation R is 0 while the error $\mu(\hat{\sigma}_\phi(Q'))$ of tuples in $\mu(\hat{\sigma}_\phi(Q'))$, where Q' is positive relational algebra over unreliable complete relations defined by maximal $\hat{\sigma}$-subexpressions $Q_1,\ldots,Q_{O(|Q|)}$ with error bound $\mu(Q_i)$ each, is bounded by $k \cdot \delta'(\epsilon_0,l) + n^k \cdot \max_i(\mu(Q_i))$. This recurrence can be solved as $k \cdot \delta'(\epsilon_0,l) \cdot \sum_{i=0}^{d} n^{k \cdot i} \leq k \cdot d \cdot \delta'(\epsilon_0,l) \cdot n^{k \cdot d}$. $\square$

It follows that there is a polynomial-time approximation algorithm for query evaluation for positive $UA[\hat{\sigma}]$ queries in the case that exact computation does not encounter singularities:

THEOREM 6.7. *Fix ϵ_0 and a positive $UA[\hat{\sigma}]$ query. There is a polynomial time algorithm that, given $0 < \delta \leq 1$, computes, for all tuples that do not have a singularity in their provenance, their membership in the result with error $\leq \delta$.*

Proof Sketch. Since $l_0 \geq 3 \cdot \log(2 \cdot k \cdot d \cdot n^{k \cdot d}/\delta)/\epsilon_0^2$ iterations of the approximation operators yield an overall approximation with error bound δ, we can proceed as follows: Start with a small value of l, say 1. Evaluate the query using that l value. Record error probabilities for each tuple while proceeding. If the error of a tuple in the output exceeds δ, double l and restart query evaluation. Repeat until the desired error bound is achieved. This is guaranteed to happen in polynomial time, at the latest when $l \geq l_0$. $\square$

7. REFERENCES

[1] L. Antova, T. Jansen, C. Koch, and D. Olteanu. "Fast and Simple Relational Processing of Uncertain Data". In *Proc. ICDE*, 2008.

[2] L. Antova, C. Koch, and D. Olteanu. "From Complete to Incomplete Information and Back". In *Proc. SIGMOD*, 2007.

[3] L. Antova, C. Koch, and D. Olteanu. "Query language support for incomplete information in the MayBMS system". In *Proc. VLDB*, 2007. Demonstration Paper.

[4] L. Antova, C. Koch, and D. Olteanu. "World-set Decompositions: Expressiveness and Efficient Algorithms". In *Proc. ICDT*, 2007.

[5] O. Benjelloun, A. Das Sarma, A. Halevy, and J. Widom. "ULDBs: Databases with Uncertainty and Lineage". In *Proc. VLDB*, 2006.

[6] J. Boulos, N. Dalvi, B. Mandhani, S. Mathur, C. Re, and D. Suciu. MYSTIQ: a system for finding more answers by using probabilities. In *Proc. SIGMOD*, 2005.

[7] N. Dalvi and D. Suciu. "Efficient query evaluation on probabilistic databases". In *Proc. VLDB*, 2004.

[8] N. Dalvi and D. Suciu. "The dichotomy of conjunctive queries on probabilistic structures". In *Proc. PODS*, 2007.

[9] M. de Rougemont. "The Reliability of Queries". In *Proc. PODS*, pages 286–291, 1995.

[10] E. Grädel, Y. Gurevich, and C. Hirsch. "The Complexity of Query Reliability". In *Proc. PODS*, pages 227–234, 1998.

[11] J. Y. Halpern. *Reasoning about Uncertainty*. MIT Press, 2003.

[12] J. M. Hellerstein, P. J. Haas, and H. J. Wang. "Online Aggregation". In *Proc. SIGMOD*, pages 171–182, 1997.

[13] C. M. Jermaine, S. Arumugam, A. Pol, and A. Dobra. "Scalable approximate query processing with the DBO engine". In *Proc. SIGMOD*, pages 725–736, 2007.

[14] R. M. Karp and M. Luby. "Monte-Carlo Algorithms for Enumeration and Reliability Problems". In *Proc. FOCS*, pages 56–64, 1983.

[15] M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.

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

[17] P. Sen and A. Deshpande. "Representing and Querying Correlated Tuples in Probabilistic Databases". In *Proc. ICDE*, pages 596–605, 2007.

[18] Stanford Trio Project. *"TriQL – The Trio Query Language"*, 2006. http://infolab.stanford.edu/∼widom/triql.html.

[19] M. Y. Vardi. "The Complexity of Relational Query Languages". In *Proc. STOC*, pages 137–146, 1982.