Answering Queries from Statistics and Probabilistic Views*

Nilesh Dalvi Dan Suciu

University of Washington, Seattle, WA, USA

1 Introduction

Systems integrating dozens of databases, in the scientific domain or in a large corporation, need to cope with a wide variety of imprecisions, such as: different representations of the same object in different sources; imperfect and noisy schema alignments; contradictory information across sources; constraint violations; or insufficient evidence to answer a given query. If standard query semantics were applied to such data, all but the most trivial queries will return an empty answer.

We believe that *probabilistic databases* are the right paradigm to model all types of imprecisions in a uniform and principled way. A probabilistic database is a probability distribution on all instances [5, 4, 15, 12, 11, 8]. Their early motivation was to model imprecisions at the tuple level: tuples are not known with certainty to belong to the database, or represent noisy measurements, etc. Tuple-independent probability distributions were sufficient for such applications, and have a very simple semantics. However, more complex types of imprecisions, like those discussed in this paper, require complex correlations between tuples, for which the query semantics has not been previously studied.

In this paper we consider two specific kinds of imprecise information, statistics on the data and explicit probabilities at the data sources. We ask a fundamental question: is it possible to answer queries by using such information ? We show that these imprecisions are modeled by a certain kind of probabilistic databases (with complex tuple correlations) and give explicit methods for answering queries, thus answering the question positively in this model. Throughout the paper we will assume the Local As View (LAV) data integration paradigm [17, 16], which consists of defining a global mediated schema $\bar{R}$, then expressing each local source i as a view $v_i(\bar{R})$ over the global schema. Users are allowed to ask queries over the global, mediated schema, $q(\bar{R})$, however the data is given as instances $J_1, \ldots, J_m$ of the local data sources. In our model all instances are probabilistic, both the local instances and the global instance. Statistics are given explicitly over the global schema $\bar{R}$, and the probabilities are given explicitly over the local sources, hence over the views. We make the Open World Assumption throughout the paper.

*Supported in part by NSF CAREER Grant IIS-0092955, IIS-0205635, and IIS-0428168, and a gift from Microsoft.

Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005

1.1 Example: Using Statistics

Suppose we integrate two sources, one showing which employee works for what department, and the second showing for each departments in which building(s) it is located.

S_1 :

name	dept
Larry Big	SalesDept
Frank Little	HR
Frank Little	SalesDept
...	...

S_2 :

dept	bldg
SalesDept	EE1
HR	EE1
HR	MGH
SalesDept	LOW
...	...

We want to find all employees working in building `EE1`. The information we have here is insufficient to answer the query, for example we cannot be certain that `Frank Little` works in the `EE1` building: he might work in the `LOW` building, or perhaps in yet another buildings (because of the Open World Assumption). Our proposal is to use statistics on the data to infer query answers with some probability. Examples of such statistics include: every department has on average 5 employees; every employee works on average in 1.2 departments; every building has about 8 departments, except `LOW` which has 20 departments; etc. Such statistics may have been collected from a different but similar data instance by using various data mining techniques, derived from other statistics

over the data, or may simply be postulated by domain experts. In addition, sometimes we also know some constraints, for example that each employee works in only one building. Our goal is to develop a general framework in which queries can be answered probabilistically from such statistics.

We formalize the problem using the LAV (local as view) method. We define a global mediated relation $R(\texttt{name}, \texttt{dept}, \texttt{bldg})$, and define mappings (views) from the global schema to the sources:

$$\begin{aligned} \texttt{S1} : v_1(n, d) &: - \quad R(n, d, -) \\ \texttt{S2} : v_2(d, b) &: - \quad R(-, d, b) \end{aligned}$$

The query is now expressed over the mediated schema:

$$q(n) \quad : - \quad R(n, -, \texttt{EE1})$$

Any statistics or constraints we know about the data are expressed over this mediated schema as well. For example, we may say:

$$\begin{aligned} & fanout_R[\texttt{dept} \Rightarrow (\texttt{name}, \texttt{bldg})] = 5 \\ & \texttt{name} \rightarrow \texttt{bldg} \end{aligned}$$

The first is a statistics, saying that the expected average number of (employee, building) pairs per each department is 5, while the second is a hard constraint (functional dependency) saying that each employee works in only one building. In this simplified example `Frank Little` is an answer to the query with probability $\approx 1/5$ (see Example 3.4). In our approach, the system computes such a probability for each employee, and ranks the answers according to their probabilities. We have chosen for illustration a very simple example; in general, the formula for the probability is much more subtle, and we will derive it in Sec. 3.

1.2 Example: Using Probabilistic Views

The Cancer Genome Anatomy Project exposes (among many other things) associations between tags and genes, and between genes and functions[1]. Thus, a very simplified fragment of the site consists of two relations:

```
TG(tag, gene)   GF(gene, function)
```

All tuples in both tables are probabilistic. In the case of `TG` the probability of each tuple derives from inherent uncertainties in the experiments that produced the tag-gene association, while in `GF` the probability is based mostly on inconsistencies found in the literature describing the gene's function. Important for our discussion is that no tuple in `TG` or `GF` is known with certainty to belong to that table. Thus, a fragment of the data may look like:

[1]The SAGE viewer and the Gene Ontology at `http://cgap.nci.nih.gov/SAGE`.

TG

Tag	Gene	P
TCCTGTAGCC	GSTA2	0.8
...	...	

GF

G	F	P
GSTA2	motor-activity	0.3
...	...	

A typical query is: find all functions that are believed to be associated to the tag `TCCTGTAGCC`. This can be expressed over a global mediated schema `R(tag,gene,function)` as:

```
q(f) :- R('TCCTGTAGCC', g, f)
```

As before, we need to use statistics to derive answers. In this particular domain, several statistics are considered common knowledge: for example, it is known that each gene has a limited number of tags (say, around 8-10), and a limited number of functions (say 4-6). However, unlike the first example, here the data at the sources is probabilistic, which further complicates the computation of the answer probabilities. We will present a general method for doing so in Sec. 4.

1.3 Summary of contributions

The paper proposes a probabilistic model for answering queries from statistics and probabilistic views. The model extends the Local As View (LAV) data integration paradigm [17, 16], by adding statistics on the global schema and probabilities to the views, and by computing probabilistic answers to queries. This is a radical departure from traditional query answering and processing techniques, where all answers are deterministic. The paper shows that probabilistic answers *can* be computed, and identifies some cases when they can be computed efficiently. Specifically, it makes the following contributions:

- It describes a model for probabilistic databases over statistics and probabilistic views; Sec. 2.
- It describes an algorithm for computing the probability of a query from statistics and (deterministic) views; Sec. 3.
- It describes an algorithm for computing the probability of a query from statistics and probabilistic views, Sec. 4.
- It describes some sufficient conditions under which the probabilistic answers to a query can be computed directly by a rewritten query Sec. 5.

2 Problem Definition

We define here our probabilistic model and the query answering problem. The model borrows ideas from probabilistic databases [12] and models of belief [3],

and adds statistics, constraints, and probabilistic views.

2.1 Preliminary Definitions

Basic Notations D denotes the finite domain of atomic values, and its cardinality is $n = |D|$. One should think of n as a very large number, say 2^{32} if $D = \texttt{int}$. $R_1, \ldots, R_k$ denote the relation names in the relational schema, and $Attr(R_i)$ is the set of attributes of R_i. $Tup(R_i)$ is the set of all possible tuples over relation R_i, and $Tup = \bigcup_{i=1,k} Tup(R_i)$ is the set of all tuples. A data instance I is a set of tuples, $I \subseteq Tup$ and $R_1^I, \ldots, R_m^I$ denotes its relational instances. We write $Inst$ $(= \mathcal{P}(Tup))$ for the set of all instances.

Unless otherwise mentioned, all queries are *conjunctive queries* [1]; occasionally, we will also consider unions of conjunctive queries and recursive datalog programs. We denote queries and views with $q, q', v, v', \ldots$, and denote boolean queries and views with upper case letters $Q, Q', V, V', \ldots$ We use $a, b, c, \ldots$ for constants, and $x, y, z, u, v, \ldots$ for variables in a query's body.

Consider the statement $t_1 \in v_1, \ \ldots, \ t_m \in v_m$, where $t_1, \ldots t_m$ are tuples and $v_1, \ldots, v_m$ are views. If $J = \{t_1, \ldots, t_m\}$, then the statement says that each tuple in J is an answer to some view (which is precisely the Open World Assumption). We will always represent such a statement as a single boolean view (query) V. For example, the single statement $t \in v$ is equivalent to the boolean conjunctive query $V = v[t/\bar{x}]$, where $\bar{x}$ are the head variables in v, and, when we have m statements, we take the conjunction of these boolean views. To illustrate, consider the view: $v(x, y) \leftarrow R(x, a, z), S(z, y)$ and $J = \{(a, b), (c, b)\}$, then the statement $(a, b) \in v, (c, b) \in v$ is equivalent to: $V \leftarrow R(a, a, z_1), S(z_1, b), R(c, a, z_2), S(z_2, b)$.

Probabilistic Databases

Definition 2.1 *A* probabilistic database *is a probability distribution on Inst, i.e.* $\mathbf{P} : Inst \rightarrow [0, 1]$ *s.t.* $\sum_I \mathbf{P}(I) = 1$. *Its entropy is:*

$$H = \sum_{I \in Inst} \mathbf{P}[I] \log \frac{1}{\mathbf{P}[I]} \quad (1)$$

We will use the terms probabilistic database and distribution interchangeably in the sequel. If P is a property on instances and f a numeric function, then P's probability and f's expected value are:

$$\mathbf{P}[P] = \sum_{I|P(I)=true} \mathbf{P}[I] \quad (2)$$

$$E[f] = \sum_I f(I)\mathbf{P}[I] \quad (3)$$

The conditional probability and the conditional expected value are given by:

$$\mathbf{P}[P_0|P] = \frac{\mathbf{P}[P_0 P]}{\mathbf{P}[P]}$$

$$E[f|P] = \frac{E[c_P f]}{\mathbf{P}[P]}$$

where $P_0 P = P_0 \wedge P$ and $c_P(I) = 1$ when $P(I) = true$, $c_P(I) = 0$ when $P(I) = false$. In this paper we are concerned with the probabilities and conditional probabilities of boolean conjunctive queries and/or of constraints.

Probabilistic Views

Given a number $p \in [0, 1]$, a *probabilistic fact* is a statement of the form $\mathbf{P}[t \in v] = p$, where v is a view and t a tuple. Equivalently, it is a statement of the form $\mathbf{P}[V] = p$, where V is a boolean view, hence we also call it a *probabilistic view*. For illustration, Sec. 1.2 showed two probabilistic facts, with probabilities 0.8 and 0.3 respectively. When $p = 1$ then we call it a *deterministic* fact or view. We denote with F a set of probabilistic facts (including any deterministic facts), and write $\mathbf{P} \models F$ if all the probabilistic facts in F hold[2] in $\mathbf{P}$. When F consists only of deterministic facts, then we express it as one single boolean view V.

Constraints

We will consider two kinds of constraints: functional dependencies (FD) and inclusion/equality constraints (IND). A functional dependency on a table R is denoted $\bar{A} \rightarrow \bar{B}$, where $\bar{A}$ and $\bar{B}$ are sets of attributes. An inclusion/equality dependency is an expression of the form $R.\bar{A} = S.\bar{B}$ or $R.\bar{A} \subseteq S.\bar{B}$. We make the restriction that every time a relation occurs in an inclusion/equality dependency it does so with the same set of attribute; e.g. we allow $R.A \subseteq S.B, S.B \subseteq T.D$, but disallow $R.A \subseteq S.B, S.C \subseteq T.D$. Hence, our dependencies are acyclic, since cyclic inclusions become equalities[3]. We write Γ for the set of FDs and INDs; $I \models \Gamma$ means that the instance I satisfies Γ; $\mathbf{P} \models \Gamma$ means that the probabilistic database $\mathbf{P}$ satisfies Γ, i.e. $\forall I.\mathbf{P}(I) > 0 \Rightarrow I \models \Gamma$; equivalently, $\mathbf{P}[\Gamma] = 1$.

Statistics

We consider two kinds of statistics in this paper, cardinalities and fan-outs, written as:

$$card_R[\bar{B}] = \sigma \quad (\sigma > 0) \quad (4)$$

$$fanout_R[\bar{A} \Rightarrow \bar{B}] = \sigma \quad (\sigma > 1) \quad (5)$$

A cardinality statistics on a relation R is written as (4) above, and states that the expected number of distinct

[2] $\mathbf{P}[V] = p$ holds, if $\mathbf{P}[V]$, when computed using Eq.(2), is p.
[3] Thus, we avoid the intractability problems due to cycles [1].

tuples in the $\bar{B}$ attributes of R is σ. More precisely, we say that a probability distribution $\mathbf{P}$ satisfies this statistics if $E[card(\Pi_{\bar{B}}(R^I))] = \sigma$. When $\bar{B} = Attr(R)$ then the statistics simply asserts the expected size of R and we write it $card(R) = \sigma$. A fanout statistics is written like (5) above, and its meaning is the following. For an instance I and $\bar{a} \in \Pi_{\bar{A}}(R^I)$, the fanout of R^I at $\bar{a}$ is the number of tuples of the form $(\bar{a}, \bar{b})$ in R^I. We say that $\mathbf{P}$ satisfies the statistics $fanout_R[\bar{A} \Rightarrow \bar{B}] = \sigma$, if $\forall \bar{a}$, the expected value of the fanout at $\bar{a}$, over all instances that contain $\bar{a}$, is σ: $E[card(\Pi_{\bar{B}}\sigma_{\bar{A}=\bar{a}}(R^I)) \mid a \in \Pi_{\bar{A}}(R^I)] = \sigma$.

We denote Σ a set of statistics, both cardinality and fanout statistics. We write $\mathbf{P} \models \Sigma$ if $\mathbf{P}$ satisfies all statistics in Σ. We restrict our model to statistics that are "chains" (but see Sec. 2.3). More precisely, we require Σ to contain precisely the following statistics about R, and no others:

$$\begin{aligned} card_R[\bar{A}_1] &= \sigma_1 > 0 \qquad (6) \\ fanout_R[\cup_{j<i}\bar{A}_j \Rightarrow \bar{A}_i] &= \sigma_i > 1, \quad i = 2, \ldots, k \end{aligned}$$

where $k \geq 1$, and $\bar{A}_1 \cup \ldots \cup \bar{A}_k$ is a partition of $Attr(R)$. One can check that the expected size of R is $\prod_{i=1}^{k} \sigma_i$.

As a simple example, consider a table $R(E, D, B)$ (similar to the example in Sec. 1.1) and the statistics:

$$\begin{aligned} card_R[D] &= \sigma_1 = 160 \\ fanout_R[D \Rightarrow E, B] &= \sigma_2 = 5 \end{aligned}$$

The expected size of R is $\sigma_1\sigma_2 = 160{*}5 = 800$.

Constraints and fanout statistics may conflict. For example $A \rightarrow B$ and $fanout[A \Rightarrow B] = 2$ are inconsistent. To eliminate such cases, we require that whenever we have an FD where A occurs on the left and B on the right, and $A \in \bar{A}_i$, $B \in \bar{A}_j$ then $i \geq j$. Similarly, whenever $R.\bar{A}$ occurs in an inclusion or equality constraint, we require $\exists i$ s.t. $\bar{A} \subseteq A_i$, and that all inclusion/equality constraints are consistent with the statistics: an equality constraint must correspond to an equality between the corresponding statistics, while an inclusion constraint must correspond to inequality.

2.2 The Problem

We will now state formally our problem. We are given the constraints Γ, statistics Σ, and probabilistic views F. Call a distribution $\mathbf{P}$ *consistent* if:

$$\mathbf{P} \models \Gamma, \quad \mathbf{P} \models \Sigma, \quad \mathbf{P} \models F$$

In general, there are many consistent distributions. To choose one, we apply the principle of indifference in probability theory, which translates into choosing the distribution that maximizes the entropy. More precisely, denote $\mathbf{P}_{\Gamma,\Sigma,F}$ that consistent distribution that has the maximum entropy H (see Eq.(1)). The problem is: given a boolean query Q, compute:

$$\mu_{\Gamma,\Sigma,F}[Q] = \lim_{n\to\infty} \mathbf{P}_{\Gamma,\Sigma,F}[Q]$$

As a variation, we are given a non-boolean query q, and want to return the set of pairs $(t, \mu_{\Gamma,\Sigma,F}[t \in v])$ where $\mu_{\Gamma,\Sigma,F}[t \in v] > 0$.

Example 2.2 We illustrate on a very simple example. Let $R(A, B)$ be a binary relation, and one single cardinality statistics $card(R) = \sigma$. Consider the following boolean query and view:

$$\begin{aligned} V &= R(a, -), R(-, b) \\ Q &= R(a, b) \end{aligned}$$

Here a, b denote constants, while $-$ denotes an anonymous variable. We want to compute $\mathbf{P}_{\Sigma,V}[Q]$ (the view here is deterministic). Intuitively this means: given that a occurs in the first column of R and b occurs in the second column, and given that the expected size of R is σ, what is the probability that the tuple (a, b) occurs in R ? Notice that we must have $n^2 \geq \sigma$ (otherwise the domain is too small R cannot have cardinality σ), and when $n^2 = \sigma$ then R contains with certainty all tuples in the domain, including (a, b), hence $\mathbf{P}[Q] = 1$. For this problem to make sense, we need to have $n^2 \gg \sigma$. We will avoid the technical complications arising from maximizing the entropy, we will consider a simple binomial distribution $\mathbf{P}$ instead. Each tuple $t \in D^2$ is inserted in R independently, with probability $p = \sigma/n^2$: the expected cardinality of R is indeed σ. Our goal now is to compute $\mathbf{P}[Q \mid V] = \mathbf{P}[QV]/\mathbf{P}[V]$ (we show later that this is $\approx \mathbf{P}_{\Sigma,V}[Q]$). It is easy to see that $\mathbf{P}[Q] = \sigma/n^2 = \mathbf{P}[QV]$ (since $Q \equiv QV$), but $\mathbf{P}[V]$ seems harder. Yet a brute force approach yields:

$$\mathbf{P}[V] = 1 - (1-p)[1 - (1 - (1-p)^{n-1})^2]$$

The resulting expression for $\mathbf{P}[Q \mid V]$ is too complex for practical use. Our approach is to let $n \to \infty$. Then $\mathbf{P}[V]$ simplifies to $(\sigma + \sigma^2)/n^2 + O(1/n^3)$, hence:

$$\mu[Q \mid V] = \lim_{n\to\infty} \mathbf{P}[Q \mid V] = 1/(1+\sigma)$$

Now it makes sense: when the domain is large, the probability of (a, b) belonging to R is about $1/(1+\sigma)$. This paper shows how to derive expressions for μ in the general case, and it also explains the relationship to the entropy maximization distribution.

2.3 Other Statistics

Despite some restrictions, our model is quite powerful, and can be used to express some complex statistics. We illustrate here through examples.

Non-chain statistics Consider the schema `R(emp,dept,bldg)` and the statistics:

$$\begin{aligned} card_R[\texttt{emp}] &= \sigma_1 \\ fanout_R[\texttt{emp} \Rightarrow \texttt{dept}] &= \sigma_2 \\ fanout_R[\texttt{dept} \Rightarrow \texttt{emp}] &= \sigma_3 \\ fanout_R[\texttt{bldg} \Rightarrow \texttt{dept}] &= \sigma_4 \end{aligned}$$

These do not form a chain, but can still be expressed in our model. First, it is easy to derive the following:

$$\begin{aligned} card_R[\texttt{dept}] &= \sigma_1\sigma_2/\sigma_3 \\ card_R[\texttt{bldg}] &= \sigma_1\sigma_2/(\sigma_3\sigma_4) \end{aligned}$$

Next, we use two new relation names: `S(ed, emp)` and `T(ed, dept, bldg)`, where `ed` represents, intuitively, `(emp, dept)` pairs, on which we define the following chain statistics and equality constraint:

$$\begin{aligned} card_S[\texttt{emp}] &= \sigma_1 \\ fanout_S[\texttt{emp} \Rightarrow \texttt{ed}] &= \sigma_2 \\ card_T[\texttt{bldg}] &= \frac{\sigma_1\sigma_2}{\sigma_3\sigma_4} \\ fanout_T[\texttt{bldg} \Rightarrow \texttt{dept}] &= \sigma_4 \\ fanout_T[\texttt{dept} \Rightarrow \texttt{ed}] &= \sigma_3 \\ S.\texttt{ed} &= T.\texttt{ed} \end{aligned}$$

This is now in our model. Finally, replace $R(x, y, z)$ with $S(u, x), T(u, y, z)$ in all queries/views.

Histograms Consider the following, listing the expected number of occurrences of departments in the table `R(Emp, Dept, Bldg)`:

Histogram

Dept	expected count
SalesDept	50
R&D	20
any other	8

Let σ be the expected number of departments in R. To express it in our model we partition R horizontally into three tables, according to their `Dept` attribute, and define these statistics:

$\texttt{R}_1\texttt{(Emp, Bldg)}$ $\quad card[R_1] = 50$
$\texttt{R}_2\texttt{(Emp, Bldg)}$ $\quad card[R_2] = 20$
$\texttt{R}_3\texttt{(Emp, Dept, Bldg)}$ $\quad card_{R_3}[\texttt{Dept}] = \sigma - 2$
$fanout_{R_3}[\texttt{Dept} \Rightarrow \texttt{Emp}, \texttt{Bldg}] = 8$

Finally, we rewrite any conjunctive query over R into a union of conjunctive queries over R_1, R_2, R_3 (see Sec. 3.1.4).

Other types Suppose 70% of `name`'s in `R(name, age)` occur in `S(name, phone)`. We express this by introducing a new table `RS(name)`, setting $card(\texttt{RS}) = 0.7 \cdot card_R[\texttt{name}]$, and defining the INDs $\texttt{RS.name} \subseteq \texttt{R.name}$, $\texttt{RS.name} \subseteq \texttt{S.name}$.

2.4 Discussion

Insufficient statistics If we lack any statistics, then a probabilistic analysis may become meaningless. For example, if we know nothing about some table R, then, by the principle of indifference, every tuple in the domain belongs to R with probability 0.5. This leads to an astronomically large expected cardinality for R; moreover, any useful evidence we may obtain from views or other statistics leads to only slight variation of the default probability 0.5, rendering them useless. In our model we insist that each table be "covered" by statistics. When none are available, some default cardinality estimates should be used.

Errors in collecting statistics Statistics are collected through data mining techniques, inferred from other statistics, or simply postulated by domain experts. In all cases one should expect to have errors. We will do an error analysis in Sec. 3.1.4 to see how sensitive the query probabilities are to errors in the statistics. The main role of query probabilities is to rank query results, so small errors may be tolerated in practice.

Tuple correlations The probability distributions we study have complex tuple correlations, which are introduced by the complex statistics, constraints, and probabilistic facts that we allow in the model. Our analysis in the paper is done for distributions with complex tuple correlations. The technical tools we deploy is to start from simpler tuple-independent distributions, then perturb them to handle correlations, but our goal is to analyze a complex, tuple-correlated distribution.

3 Using Statistics

We have a set of statistics Σ, a set of constraints Γ, and one (deterministic) boolean view V. We show here how to compute the limit probability, $\mu_{\Gamma,\Sigma,V}[Q]$, for a conjunctive boolean query Q. We proceed in two steps. First we study the probabilities of queries under a specific binomial distribution $\mathbf{P}$ based on Σ, and will show how to compute the conditional probability $\mathbf{P}[Q \mid V, \Gamma]$, and its limit $\mu[Q \mid V, \Gamma]$ for the binomial distribution. This is the hardest technical result in this paper. Then we show that this is a close approximation of the entropy-maximizing distribution $\mathbf{P}_{\Gamma,\Sigma,V}$.

3.1 The Binomial Distribution

The *binomial distribution* $\mathbf{P}$ introduced here is associated to a set of statistics Σ, which we assume fixed.

3.1.1 Definition

Consider a single relation $R(A_1, \ldots, A_m)$ with m attributes, and let us start by assuming a single cardinality statistics on R, $card(R) = \sigma$. The associated binomial distribution is: each tuple in D^m has probability $p = \sigma/n^m$. Tuples in R are chosen independently and with probability p. Hence, the binomial distribution is $\mathbf{P}[I] = p^{|I|}(1-p)^{n^m - |I|}$.

The probability $\mathbf{P}[I \neq \emptyset]$ is $1 - (1 - \sigma/n^m)^{n^m}$, and the expected cardinality of a nonempty I is $\sigma/(1 - (1 - \sigma/n^m)^{n^m})$, i.e. slightly larger than σ. We need below a binomial distribution for which the expected cardinality of a nonempty I is exactly σ. This is precisely the binomial distribution for the statistics

$\hat{\sigma}$, s.t. $\hat{\sigma}/(1-(1-\hat{\sigma}/n^m)^{n^m}) = \sigma$. Such a $\hat{\sigma}$ exists and is unique if and only if $\sigma > 1$.

Consider now an arbitrary set of statistics on R, and we use the notations in Eq.(6) in Sec. 2. Denote $\bar{B}_i = \bigcup_{j \le i} A_j$. We define the following distribution, which we still call "binomial". Let $m_i = |\bar{B}_i|$, and $R^{(i)} = \Pi_{\bar{B}_i}[R]$, for $i = 1, \ldots, k$. The generative model starts by choosing randomly an instance for $R^{(1)}$, using a binomial distribution for σ_1: i.e., the expected size of $R^{(1)}$ is σ_1. Next, for each tuple $\bar{b}_1 \in R^{(1)}$ generate a random non-empty instance of tuples $\bar{a}_2$, using binomial distribution $\hat{\sigma}_2$ ($\hat{\sigma}_2$ exists since $\sigma_2 > 1$): $R^{(2)}$ consists of all tuples $(\bar{b}_1, \bar{a}_2)$ thus generated. The expected size of $R^{(2)}$ is $\sigma_1\sigma_2$. In general, generate $R^{(i)}$ as follows: for each tuple $\bar{b}_{i-1} \in R^{(i-1)}$ generate a random nonempty instance of tuples $\bar{a}_i$ using binomial distribution $\hat{\sigma}_i$. $R^{(i)}$ consists of all tuples $(\bar{b}_{i-1}, \bar{a}_i)$. Finally, output $R = R^{(k)}$. This gives us a probability distribution $\mathbf{P}$. We can prove that $\mathbf{P}$ indeed satisfies the statistics Σ.

When the schema consists of multiple relations $R_1, \ldots, R_k$, the binomial distribution is defined independently on each relation. In the sequel, $\mathbf{P}$ denotes a binomial distribution associated to some statistics Σ.

3.1.2 Two Query Parameters

Our main result expresses $\mu[Q \mid V]$ in terms of two query parameters, called *exponent* and the *coefficient*, whose definition requires lots of notations. As a consequence this section is quite technical, and may be skipped by a reader interested only in the high level results. The important point is that these parameters are just two numbers, which can be computed from the query expression, the statistics, and the constraints, in exponential time in their sizes. To make the notations below more readable we proceed in three steps an illustrate along the way with the following running example:

`Schema`	$R(A,B,C,D), S(E,F)$
`Statistics`	$card_R[A] = \sigma_1$
	$fanout_R[A \Rightarrow BC] = \sigma_2$
	$fanout_R[BC \Rightarrow D] = \sigma_3$
	$card_S[E] = \sigma_4$
	$fanout_S[E \Rightarrow F] = \sigma_5$
`Query`	$Q :- R(a,u,v,x), R(a,x,w,y), S(y,z)$

In Q, a is a constant while x, y, z, u, v, w are variables.

From Q to Q$^{(*)}$ First we extend the schema based on the statistics. If Σ partitions the attributes of a relation R into k sets $\bar{A}_1, \ldots, \bar{A}_k$, and we denote $\bar{B}_i = \bigcup_{j \le i} \bar{A}_j$, then we introduce k new relation names: $R^{(1)}(\bar{B}_1)$, ..., $R^{(k)}(\bar{B}_k)$; we may identify $R^{(k)}$ with R, since they have the same attributes. The *proper arity* of $R^{(i)}$ is $A(R^{(i)}) = |\bar{A}_i|$ and the *proper attributes* of $R^{(i)}$ are $\bar{A}_i$. We illustrate the extended schema on our running example, and underline the proper attributes (not to be confused with keys):

$$R^{(1)}(\underline{A}), R^{(2)}(A, \underline{B,C}), R^{(3)}(A,B,C,\underline{D})$$
$$S^{(1)}(\underline{E}), S^{(2)}(E, \underline{F})$$

Given a query Q, we construct $Q^{(*)}$ by expanding each subgoal referring to R into k subgoals on the relations $R^{(1)}, \ldots, R^{(k)}$, then eliminate duplicate subgoals. In our example:

$$\begin{aligned} Q^{(*)} \quad :- \quad & R^{(1)}(\underline{a}), R^{(2)}(a, \underline{u,v}), R^{(3)}(a,u,v,\underline{x}), \\ & R^{(2)}(a, \underline{x,w}), R^{(3)}(a,x,w,\underline{y}), \\ & S^{(1)}(\underline{y}), S^{(2)}(y, \underline{z}) \end{aligned}$$

The subgoal $R^{(1)}(\underline{a})$ initially occurred twice, and we kept only one occurrence.

The arity, degree, and constant of a query Here we associate three constants to a query Q, the arity, the degree, and the constant, denoted $A(Q)$, $D(Q)$, $C(Q)$. The first two are:

$$\begin{aligned} A(Q) &= \sum_{g \in subgoals(Q^{(*)})} A(g) \qquad (7) \\ D(Q) &= A(Q) - V(Q) \end{aligned}$$

$A(g)$ denotes the proper arity of the relation occurring in the subgoal g, and $V(Q)$ is the number of variables in Q. In our running example (count the underlined attributes in $Q^{(*)}$):

$$\begin{aligned} A(Q) &= 1+2+1+2+1+1+1 = 9 \\ D(Q) &= 9-6 = 3 \end{aligned}$$

To define $C(Q)$ we need more definitions. An occurrence of a variable in a subgoal of $Q^{(*)}$ is called *proper* if it is in a proper attribute; a variable in $Q^{(*)}$ is *trivial* if it has only one proper occurrence; and a subgoal in $Q^{(*)}$ is called *trivial* if all its proper attributes have trivial variables. In our running examples u, v, w, z are trivial variables (since they are underlined only once in $Q^{(*)}$), and $R^{(2)}(a, \underline{u,v})$, $S^{(2)}(y, \underline{z})$ are trivial subgoal (since their proper attributes are all trivial variables); all other subgoals are non-trivial.

$C(Q)$ is given by a product, consisting of one factor $C_{nt}(g)$ for each non-trivial subgoal g, and one factor $C_t(R^{(i)})$ for *all* trivial subgoals of type $R^{(i)}$. If g is a non-trivial subgoal referring to relation $R^{(i)}$, then:

$$C_{nt}(g) = \begin{cases} \sigma_i/(1-e^{-\sigma_{i+1}}) & i \le k \\ \sigma_i & i = k \end{cases} \qquad (8)$$

If there are l trivial subgoals for $R^{(i)}$, then:

$$C_t(R^{(i)}) = 1 - e^{-\sigma_i} \sum_{0 \le j \le l} \frac{(\sigma_i)^j}{j!} \qquad (9)$$

This number is 1 when $l = 0$. Finally, $C(Q)$ is:

$$C(Q) = \prod_{g \in \text{non-trivial-subgoals}(Q^{(*)})} C_{nt}(g) \prod_{R^{(i)}} C_t(R^{(i)})$$

For our running example, $C(Q)$ is given below. The first line represent $C_{nt}(g)$ for the five non-trivial subgoals, the second line represents $C_t(R^{(2)})$ and $C_t(S^{(2)})$, each of which has $l = 1$ trivial subgoal.

$$\begin{aligned} C(Q) &= \frac{\sigma_1}{1-e^{-\sigma_2}}\sigma_3\frac{\sigma_2}{1-e^{-\sigma_3}}\sigma_3\frac{\sigma_4}{1-e^{-\sigma_5}} \\ &\quad (1-e^{-\sigma_2})(1-e^{-\sigma_5}) \\ &= \frac{\sigma_1\sigma_2\sigma_3^2\sigma_4}{1-e^{-\sigma_3}} \end{aligned}$$

The exponent and the coefficient Finally, we will define the exponent and the coefficient of Q. Now we will take the constraints Γ into account, and will start by "chasing" Q with all inclusion/equality dependencies in Γ: chasing with the dependency $R.\bar{A} \subseteq S.\bar{B}$ means adding a new subgoal $S(-,\ldots,-,\bar{x},-,\ldots,-)$ for every subgoal $R(\ldots,\bar{x},\ldots)$ in Q; chasing for an equality dependency means chasing for the inclusions in both directions. Since there are no cycles, this process terminates. After chasing, we minimize the query. Hence, in the sequel, we will assume that Q is chased and minimized.

Next we consider substitutions of the variables in Q with variables and/or constants: a substitution is not allowed to use other constants except those already present in Q. We will consider all possible substitutions on Q, denoting $h(Q)$ for the query obtained by applying the substitution h to Q. We do not distinguish between isomorphic queries (which can be transformed into the other by renaming variables), hence it suffices to consider only substitutions that use only the variables in Q, and therefore we need to consider only exponentially many substitutions h. We write $h(Q) \models \Gamma$ if $h(Q)$ viewed as a canonical database satisfies[4] Γ.

A substitution h partitions the subgoals of Q into equivalence classes, s.t. g and g' are in the same equivalence class if $h(g) = h(g')$. We say that h is a *most general unifier* if for any other substitution h' producing the same partition as h, there exists f s.t. $h' = f \circ h$. We will consider in the sequel only substitutions h that are most general unifiers, and in this case call $G = h(Q)$ a *most general unifying query* for Q. For example, assume a ternary table $R(A,B,C)$ and the query $Q = R(a,x,y), R(z,b,b)$. Assume a cardinality constraint on R, i.e. $k = 1$, hence $Q^{(*)} = Q$. There are exactly two most general unifying queries: Q itself and $G = R(a,b,b)$; the query $G' = R(a,x,b), R(z,b,b)$ is not most general unifying. Now suppose that we have a cardinality statistics on C and a fanout statistics $C \Rightarrow A,B$. Then $Q^{(*)} = R^{(1)}(y), R(a,x,y), R^{(1)}(b), R(z,b,b)$, and we are allowed to "unify" y and b, hence the most general unifying queries are now Q, G, and G'.

We can finally define the query's exponent *exp*(Q) and coefficient *coeff*(Q):

$$\begin{aligned} MGUQ_\Gamma(Q) &= \{h(Q) \mid h = \text{most general unif.}, \\ &\quad h(Q) \models \Gamma\} \\ exp_\Gamma(Q) &= \min\{D(G) \mid G \in MGUQ_\Gamma(Q)\} \\ MGUQ^0_\Gamma(Q) &= \{G \mid G \in MGUQ_\Gamma(Q), \\ &\quad D(G) = exp_\Gamma(Q)\} \\ coeff_\Gamma(Q) &= \sum\{C(G) \mid G \in MGUQ^0_\Gamma(Q)\} \end{aligned}$$

$MGUQ_\Gamma(Q)$ is the set of all most general unifying queries, and contains at most exponentially many queries; hence both numbers $exp_\Gamma(Q)$ and $coeff_\Gamma(Q)$ can be computed in exponential time. We will drop the subscript Γ when $\Gamma = \emptyset$, and write $MGUQ(Q)$ etc.

In our running example, $MGUQ(Q) = \{Q, G_1, G_2\}$ where:

$$\begin{aligned} G_1 &:- R(a,x,w,x), R(a,x,w,y), S(y,z) \\ G_2 &:- R(a,x,w,x), S(y,z) \end{aligned}$$

G_1 is obtained by unifying the two $R^{(2)}$ subgoals in $Q^{(*)}$ (hence $u = x$, $v = w$), while G_2 is obtained by unifying these two, plus the two $R^{(3)}$ subgoals (hence $u = x$, $v = w$, $x = y$). We have $exp(Q) = A(Q) = A(G_1) = A(G_2) = 3$, hence $MGUQ^0(Q) = \{Q, G_1, G_2\}$ and $coeff(Q) = C(Q) + C(G_1) + C(G_2)$. We have seen $C(Q)$ already; $C(G_1)$ and $C(G_2)$ are computed similarly and give: $C(G_1) = \frac{\sigma_1\sigma_2\sigma_3^2\sigma_4}{(1-e^{-\sigma_2})(1-e^{-\sigma_3})}$, $C(G_2) = \frac{\sigma_1\sigma_3\sigma_4}{(1-e^{-\sigma_2})}$.

3.1.3 Query Probability

Here we state our technically most difficult result: how to compute the query probability $\mu[Q \mid V, \Gamma]$ for the binomial distribution associated to a set of statistics Σ. This, in essences, solves our goal of answering a query from a set of statistics (since we will show in the following section that this distribution is the same, for practical purposes, as the entropy-maximization distribution). The expression of μ will use the exponent and the coefficient introduced in the previous section. All proofs are omitted, and can be found in [7].

Theorem 3.1 *Let Σ be a set of statistics, $\mathbf{P}$ the binomial distribution for Σ, and Γ a set of constraints. Let Q be a conjunctive query. Then:*

$$\mathbf{P}[Q|\Gamma] = \frac{\text{coeff}_\Gamma(Q)}{n^{\text{exp}_\Gamma(Q)}} + O(\frac{1}{n^{\text{exp}_\Gamma(Q)+1}})$$

[4] $h(Q)$ always satisfies the IND's; we only have to check if it satisfies the FDs.

Corollary 3.2

$$\mu[Q \mid V, \Gamma] \quad = \quad \begin{cases} \frac{\text{coeff}_\Gamma(QV)}{\text{coeff}_\Gamma(Q)} & \text{if } \exp_\Gamma(QV) = \exp_\Gamma(V) \\ 0 & \text{if } \exp_\Gamma(QV) > \exp_\Gamma(V) \end{cases}$$

We illustrate the results with several examples.

Example 3.3 Continuing Example 2.2, $MGUQ(V) = \{Q, V\}$ and $MGUQ(Q) = \{Q\}$. We have $D(V) = D(Q) = 2$, $MGUQ^0(V) = MGUQ(V)$, $C(V) = \sigma^2$, $C(Q) = \sigma$. Hence $exp(V) = 2$, $coeff(V) = \sigma + \sigma^2$. Similarly we can compute $exp(QV) = 2$, $coeff(QV) = \sigma$ (since $MGUQ^0(QV) = \{Q\}$). The theorem and corollary give us:

$$\begin{aligned} \mathbf{P}[V] &= \frac{\sigma^2 + \sigma}{n^2} + O(\frac{1}{n^3}) \\ \mathbf{P}[QV] &= \frac{\sigma}{n^2} + O(\frac{1}{n^3}) \\ \mu[Q \mid V] &= \frac{1}{1+\sigma} \end{aligned}$$

Example 3.4 This example is adapted from the motivating example in Sec. 1.1. We have one relation $R(N, D, B)$ with statistics: $card_R(D) = \sigma_1$, $fanout_R(D \Rightarrow N, B) = \sigma_2$. The view and the query are:

$$\begin{aligned} V \quad :- \quad & R(\texttt{LarryBig}, \texttt{SalesDept}, -), \\ & R(-, \texttt{SalesDept}, \texttt{EE1}) \\ Q \quad :- \quad & R(\texttt{LarryBig}, -, \texttt{EE1}) \end{aligned}$$

In other words, we know that `LarryBig` works in the `SalesDept` and that the `SalesDept` is in building `EE1` and want to find the probability that `LarryBig` is in building `EE1`. Start by computing $V^{(*)}$:

$$\begin{aligned} V^{(*)} \quad :- \quad & R^{(1)}(\texttt{SalesDept}), \\ & R(\texttt{LarryBig}, \texttt{SalesDept}, -), \\ & R(-, \texttt{SalesDept}, \texttt{EE1}) \end{aligned}$$

We have: $D(V) = 5 - 2 = 3$, $C(V) = \sigma_1\sigma_2^2/(1 - e^{-\sigma_2})$. $MGUQ(V)$ contains two queries, namely V itself and W :- R(`LarryBig`, `SalesDept`, `EE1`), and both have $D(V) = D(W) = 3$. Hence:

$$\begin{aligned} exp(V) &= 3 \\ coeff(V) &= (\sigma_1\sigma_2^2 + \sigma_1\sigma_2)/(1 - e^{-\sigma_2}) \end{aligned}$$

Consider now $MGUQ(QV)$. Here there is a single query with degree 3, namely W above, obtained now by unifying all three subgoals in QV. Hence:

$$\begin{aligned} exp(QV) &= 3 \\ coeff(QV) &= \sigma_1\sigma_2/(1 - e^{-\sigma_2}) \end{aligned}$$

It follows that $\mu[Q|V] = 1/(1 + \sigma_2)$.

Example 3.5 Functional dependencies affect μ, as the following example illustrates. Assume $R(A, B, C, D, E)$ with cardinality statistics $card(R) = \sigma$, and consider the view and query:

$$\begin{aligned} V \quad :- \quad & R(a, b, d, f, g), \\ & R(a, -, c, f, -), R(a', -, c', f, -), \\ & R(-, b, -, f, h), R(-, b', -, f, h) \\ Q \quad :- \quad & R(-, b, c, -, -) \end{aligned}$$

Then $MGUQ^0(V) = \{V_1, V_2\}$ where:

$$\begin{aligned} V_1 \quad :- \quad & R(a, b, d, f, g), R(a, b, c, f, h), R(a', b', c', f, h) \\ V_2 \quad :- \quad & R(a, b, d, f, g), R(a, b', c, f, h), R(a', b, c', f, h) \end{aligned}$$

$D(V_1) = D(V_2) = exp(V) = 15$, and $C(V_1) = C(V_2) = \sigma^3$. Considering Q, $MGUQ^0(QV) = \{V_1\}$ and $\mu[Q \mid V] = 1/2$. If we add the FD $A \to B$, then $V_2 \not\models \Gamma$ and $MGUQ^0_\Gamma(V) = MGUQ^0_\Gamma(QV) = \{V_1\}$ and $\mu[Q \mid V, \Gamma] = 1$. In general, adding FD's may increase or decrease μ, or increase $exp(-)$. Similarly, inclusion/equality dependencies affect the probabilities: they may increase the exponent.

3.1.4 Discussion

More complex queries Our two main results extend immediately to unions of conjunctive queries. For example, assume $Q = Q_1 \cup Q_2 \cup Q_3$: we have seen in Sec. 2.4 that we need to consider such queries when handling histogram style statistics. The inclusion-exclusion principle gives us the following formula for the probability of Q:

$$\mathbf{P}[Q] \quad = \quad \sum_i \mathbf{P}[Q_i] - \sum_{i \neq j} \mathbf{P}[Q_i Q_j] + \mathbf{P}[Q_1 Q_2 Q_3]$$

Each query on the right hand side is a conjunctive query, and we can apply Theorem 3.1 to each individually. This can be used to compute $\mu[Q \mid V]$ when both Q and V are unions of conjunctive queries. Another immediate extension is to queries with the inequality predicate, $\neq$: both Theorem 3.1 and Corollary 3.2 carry over to this case[5]. More complex predicates like $x < y$ or x like y require separate treatment: as a heuristics, one can associate to them a default probability, say 0.5 to the first and 0.1 to the second.

Error analysis We have discussed that most statistics σ should be expected to have errors, and we need to understand their impact on the computed probabilities. First, we will replace all factors $(1 - e^{-\sigma})$ with 1, since they are ≈ 1: e.g. for $\sigma > 4.6$, $0.99 < 1 - e^{-\sigma} < 1$. Then, we note that $\mu[Q \mid V]$ is a fraction of two

[5] We need to adjust with the number of automorphisms of Q; e.g. $Q = R(x, y)R(y, z)R(z, x), x \neq y, y \neq z, z \neq x$, then we need to adjust with 1/3. Without inequalities, the query unifies to $R(x, x)$ and no adjustment is needed.

polynomials in the variables $\sigma_1, \sigma_2, \ldots$ (all the numbers occurring in the statistics Σ). Moreover, all coefficients are ≥ 0. We do the error analysis for one statistics σ at a time, hence the probability is $f(\sigma) = \mu[Q \mid V] = P_1(\sigma)/P_2(\sigma)$, where P_1, P_2 are polynomials in σ. Let d_1, d_2 be their degrees (obviously $d_1 \leq d_2$). An inspection of the expressions for *coeff*(QV) and *coeff*(V) shows that d_1 is bounded by the number of subgoals in $(QV)^{(*)}$ that refer to the table R, and d_2 is bounded by the number of such subgoals in $V^{(*)}$, where R is the table to which the statistics σ applies. For the error analysis, we compute the derivative[6] of f:

$$\begin{aligned} |f'(\sigma)| &= |(\frac{P_1'(\sigma)}{P_1(\sigma)} - \frac{P_2'(\sigma)}{P_2(\sigma)})|f(\sigma) \\ &\leq \frac{(d_1+d_2)}{\sigma} f(\sigma) \end{aligned}$$

which leads to the following formula for the relative error: $|\Delta f|/f \leq (d_1+d_2)|\Delta\sigma|/\sigma$. Thus, the relative error increases by at most a factor bounded by the number of subgoals in the query/view relevant to the statistics σ.

Complexity A naive algorithm for computing $\mu[Q \mid V, \Gamma]$ that applies Corollary 3.2 directly runs in exponential time in Q and V.

3.2 The Entropy Maximization Distribution

We now return to our original goal, of computing the query answer for the entropy maximization distribution: so far we have shown only how to compute the binomial distribution. This section shows how they are related (all proofs are deferred to [7]). First, one can check directly (using Lagrange multipliers) that, in the absence of constraints and views, the entropy-maximization distribution $\mathbf{P}_\Sigma$ is equal to the binomial distribution $\mathbf{P}$:

Proposition 3.6 $\mathbf{P}_\Sigma = \mathbf{P}$

Next we relate the binomial distribution to $\mathbf{P}_{\Sigma,\Gamma,V}$. We first relate $\mathbf{P}_{\Sigma,\Gamma,V}[Q]$ to $\mathbf{P}_{\Sigma,\Gamma}[Q \mid V]$, then the latter to $\mathbf{P}_\Sigma[Q \mid V, \Gamma]$, which is the binomial distribution $\mathbf{P}[Q \mid V, \Gamma]$. Since both Γ and V are boolean properties on instances, the two steps are instances of the following lemma, relating two entropy-maximization distributions:

Lemma 3.7 *Let Σ be a set of statistics, and let P_1, P_2 be any two boolean properties on instances. Then there exists a set of perturbed statistics $\hat{\Sigma}$ s.t. for any boolean query Q:*

$$\mathbf{P}_{\Sigma,P_1,P_2}[Q] = \mathbf{P}_{\hat{\Sigma},P_1}[Q \mid P_2]$$

[6]We use $\sigma_1 P_1'(\sigma) \leq d_1 P_1(\sigma)$, $\sigma_2 P_2'(\sigma) \leq d_2 P_2(\sigma)$.

We consider now the relationship between Σ and $\hat{\Sigma}$, showing that the perturbation is small, although the exact difference may be difficult to compute in practice. We consider this separately for V and for Γ.

Perturbation due to the view We will only describe here the case where Σ consists of cardinality statistics for each relation, which we denote $card(R_i) = \sigma_i$, for $i = 1, \ldots, k$. Then, in $\hat{\Sigma}$ the statistics become $card(R_i) = \hat{\sigma}_i$. Intuitively, we expect σ_i to be greater than $\hat{\sigma}_i$, roughly by the amount equal to the number of subgoals of R_i in V. The exact formula is as follows. Define:

$$G_i(G) = \text{number of subgoals in } G \text{ that refer to } R_i$$

Then:

Proposition 3.8 *For every $i = 1, \ldots, k$:*

$$\sigma_i = \hat{\sigma}_i + \frac{\sum_{G \in MGUQ^0_\Gamma(V)} G_i(G) C(G)}{\sum_{G \in MGUQ^0_\Gamma(V)} C(G)} \quad (10)$$

Notice that $0 < \sigma_i - \hat{\sigma}_i \leq G_i(V)$.

To find $\hat{\Sigma}$ one needs to solve the algebraic Equation (10), which may be difficult in general. However, since the perturbations are small, for all practical purposes one can take $\hat{\Sigma} \approx \Sigma$.

Example 3.9 Consider the query and view in Example 2.2, with the statistics $\Sigma : card(R) = \sigma$. Want to find a perturbed cardinality statistics $\hat{\Sigma} : card(R) = \hat{\sigma}$ s.t. $\mathbf{P}_{\Sigma,V}[Q] = \mathbf{P}[Q \mid V]$, where $\mathbf{P}$ is the binomial distribution for $\hat{\sigma}$. Recall that $MGUQ^0(V) = \{Q, V\}$ (see Example 3.3), hence

$$\sigma = \hat{\sigma} + \frac{2\hat{\sigma}^2 + 1\hat{\sigma}}{\hat{\sigma}^2 + \hat{\sigma}}$$

This leads to an equation of degree 2 in $\hat{\sigma}$. Since $\sigma - 2 < \hat{\sigma} < \sigma - 1$, we argue to approximate $\hat{\sigma}$ with σ in practice, rather than solve the equation.

Perturbations due to the constraints We briefly illustrate here functional dependencies, omitting inclusion/equality constraints. FDs cause even smaller perturbations, which are asymptotically 0:

Proposition 3.10 *For any statistics Σ and FDs Γ,*

$$|\frac{\mathbf{P}_{\Sigma,\Gamma}[Q]}{\mathbf{P}_\Sigma[Q|\Gamma]} - 1| \leq O(\frac{1}{n})$$

The intuition behind this is that a randomly chosen database instance almost certainly satisfies a functional dependency. This is because a functional dependency is the negation of a conjunctive query with $\neq$, and it follows from Theorem 3.1 that $1 - \mathbf{P}_\Sigma[\Gamma] \leq O(1/n)$. Thus, adding functional dependencies does not change the statistics asymptotically. However, functional dependencies *do* affect query probabilities, see Example 3.5.

4 Using Probabilistic Views

We now turn to our second major problem: how to answer a query from statistics *and* probabilistic views. We have a set of statistics Σ, a set of constraints Γ, and a set of probabilistic facts F. Our second main result in this paper shows how to answer a query Q from statistics and probabilistic views, by giving an explicit formula for the limit probability $\mu_{\Gamma,\Sigma,F}[Q]$. Recall that the set of probabilistic facts F can be represented by a set of m boolean views $V_1, \ldots, V_m$ and m probabilities $p_1, \ldots, p_m \in [0,1]$: F is the collection of statements $\mathbf{P}[V_j] = p_j$, $j = 1, m$.

To compute $\mathbf{P}_{\Gamma,\Sigma,F}[Q]$ we will express this probability in terms of a binomial distribution $\mathbf{P}_{\hat{\Sigma}}[- \mid \Gamma]$, for a slightly perturbed set of statistics $\hat{\Sigma}$. However, this step is more involved than Lemma 3.7, because the m probabilistic facts cannot be consolidated into one single view: instead we need to consider 2^m "views", representing all possible overlaps. For that we introduce the following notations: given m boolean views $V_1, \ldots, V_m$, and m constants $p_1, \ldots, p_m$, for any set $\Delta \subseteq \{1, \ldots, m\}$, denote:

$$\begin{aligned} p_\Delta &= \prod_{j \in \Delta} p_j \\ \bar{p}_\Delta &= \prod_{j \in \Delta} p_j \prod_{j \notin \Delta} (1 - p_j) \qquad (11) \\ V_\Delta &= \bigwedge_{j \in \Delta} V_j \\ \bar{V}_\Delta &= \bigwedge_{j \in \Delta} V_j \wedge \bigwedge_{j \notin \Delta} \neg V_j \end{aligned}$$

Before we can state our main result here, we note that the probabilistic facts may be conflicting. For example if we state that the probability of $V_1 : -R(a,b)$ is $p_1 = 0.5$ and the probability of $V_2 : -R(a,-)$ is $p_2 = 0.1$, then we have a contradiction, since V_1 logically implies V_2. To avoid such cases we require the views to be *non-conflicting*, which means: for any j, $\mu[V_j \mid W] = 0$, where W is the conjunction of all views other than V_j. Then, we have:

Theorem 4.1 *Assuming the views are non-conflicting, the probabilistic answer to a query given a set of statistics and probabilistic views is:*

$$\mu_{\Gamma,\Sigma,F}[Q] = \sum_{\Delta} \bar{p}_\Delta \mu_{\hat{\Sigma}}[Q \mid V_\Delta, \Gamma] \qquad (12)$$

Here $\mu_{\hat{\Sigma}}$ is the limit of the binomial distribution for a perturbed statistics $\hat{\Sigma}$; the perturbation from $\hat{\Sigma}$ to Σ is bounded by the size of F.

Here $\bar{p}_\Delta$ is given by Eq.(11), while $\mu_{\hat{\Sigma}}[Q \mid V_\Delta, \Gamma]$ is given in Corollary 3.2. The proof of the theorem is sketched in the Appendix. The theorem leads to an exponential time algorithm: we address this in Sec. 5.‘

Example 4.2 Consider our example in Sec. 1.2, for which we assume the following set of statistics Σ: $card_R(\texttt{gene}) = \sigma_1$, $fanout_R(\texttt{gene} \Rightarrow \texttt{tag}, \texttt{function}) = \sigma_2$. We abbreviate the constants `TCCTGTAGCC`, `GSTA2`, and `motor-activity` with t, g, and m respectively, hence we have:

$$\begin{aligned} V_1 &: - \quad R(t,g,-) \quad p_1 = 0.8 \\ V_2 &: - \quad R(-,g,m) \quad p_2 = 0.3 \\ Q &: - \quad R(t,-,m) \end{aligned}$$

The views are indeed non-conflicting: $\mu_\Sigma[V_1 \mid V_2] = \mu_\Sigma[V_2 \mid V_1] = 0$. We apply Eq.(12) on all three nonempty sets Δ : $\mu_\Sigma[Q \mid V_1] = \mu_\Sigma[Q \mid V_2] = 0$, $\mu_\Sigma[Q \mid V_1, V_2] = 1/(1+\sigma_2)$. It follows that $\mu_{\Sigma,F}[Q] = p_1 p_2/(1+\sigma_2)$: we make a small error here by using σ_2 instead of the perturbed statistics $\hat{\sigma}_2$: the latter is difficult to compute, and differs by at most 2 (size of F) from σ_2.

5 Query Rewriting

We now address how to evaluate queries from statistics and probabilistic views *efficiently*. The method we consider here consists of rewriting the query in terms of the view instances; other possibilities exists, such as approximations through Monte Carlo simulations, but they are beyond the scope of this paper. In [7] we give two sufficient conditions under which such rewritings are possible; for lack of space, we do not include here the technical conditions, but only illustrate with two examples, then give a general result for probabilistic views.

Let $\bar{v} = v_1, \ldots, v_m$ be a set of (non-boolean) views and $J = J_1, \ldots, J_m$ an instance for these views. We denote with $\bar{v}[J]$ the boolean view that is the conjunction of all expressions of the form $v_i[t]$, for $t \in J_i$, $i = 1, m$ (see Sec 2). Given a (non-boolean) query q we say that a tuple t is an **almost certain answer** if $\mu_{\Gamma,\Sigma,\bar{v}_J}[t \in q] = 1$; we say that t is a **probable answer** if $\mu_{\Gamma,\Sigma,\bar{v}_J}[t \in q] > 0$. Two technical definitions in [7] give sufficient conditions under which the set of almost-certain, or the set of probable answers can be computed by a rewritten query, q_r, evaluated on J. We illustrate with two examples.

Example 5.1 Consider the view and query below:

$$\begin{aligned} v(x,y) &: - \quad R(x,z), R(y,z) \\ q(x,y) &: - \quad R(x,z), R(y,z) \end{aligned}$$

A view instance J can be thought of as a graph, consisting of edges $v(m,n)$. Then, a tuple (a,b) is a probable answers iff it is an almost certain answer, iff (a,b) is in the transitive closure of J. Hence, the set of probable answers can be computed by the following recursive datalog program q_r:

$$\begin{aligned} q_r(x,y) &: - \quad v(x,y) \\ q_r(x,y) &: - \quad v(x,z), q_r(z,y) \end{aligned}$$

To see an illustration, consider the following instance for v: $J = \{(m,n),(n,p),(r,s)\}$. Then the boolean query $v[J]$ is:

$$V_J \quad :- \quad R(m,z_1), R(n,z_1), R(n,z_2), \\ R(p,z_2), R(r,z_3), R(s,z_3)$$

and its unique unifier with minimum D is:

$$U \quad :- \quad R(m,z), R(n,z), R(p,z), R(r,z_3), R(s,z_3)$$

U encodes the connected components of J: the distinct variables represent the connected components, and a subgoal $R(n,x)$ in U means that the node n belongs to the component x. One can check now that for any two nodes a, b, $\mu[q(a,b) \mid v[J]]$ is 1 when a, b, and is 0 when a, b are not connected.

Example 5.2 Consider v_1, v_2, q from the example in Sec. 1.1, and assume one cardinality statistics $card(R) = \sigma$. Then all probable answers are computed by:

$$q_r(n) \quad :- \quad v_1(n,d), v_2(d, \texttt{EE1})$$

This is a simple join query, which needs to be evaluated on the two tables S_1 and S_2. To appreciate the importance of such a rewriting, it helps reviewing the direct approach that applies Corollary 3.2 naively: first build a huge boolean view V consisting of all conjunction of views $v_1[t]$, $t \in S_1$ and $v_2[t]$, $t \in S_2$, then iterate over all tuples t in the active domain and compute the boolean query $Q = q[t]$; then, apply Corollary 3.2 to Q and V; t is probable iff $exp(QV) = exp(V)$.

Finally, we show how to compute efficiently query answers from statistics and probabilistic views. Now J is a probabilistic view instance, i.e. a set of tuples with probabilities, and $\bar{v}[J]$ denotes the corresponding set of probabilistic facts. We want to compute for each tuple t, the probability $\mu_{\Gamma,\Sigma,\bar{v}[J]}[t \in q]$; equivalently, compute the set of pairs $(t, \mu_{\Gamma,\Sigma,\bar{v}[J]}[t \in q])$. For that we will evaluate a rewritten query q_r on the probabilistic database J. This is a non-standard query evaluation, but efficient methods exists for evaluating queries under the probabilistic semantics [8]. We have:

Proposition 5.3 *Let $\bar{v} = v_1, \ldots, v_m$ be a set of views. Suppose that q admits a rewriting q_r that computes the almost-certain answers. Then, if q_r is evaluated on the probabilistic instance J, it computes precisely the set $(t, \mu_{\Gamma,\Sigma,\bar{v}[J]}[t \in q])$.*

6 Related Work

Several models of probabilistic databases [5, 4, 15, 12, 11] have been proposed in the past that represent uncertainties at tuple level. In our recent work [8], we give efficient algorithms for evaluation of SQL queries on such databases.

In [6] we have obtained some preliminary results on asymptotic conditional query probability, without statistics, constraints or probabilistic views.

There is a lot of work on using statistics and subjective information in knowledge bases. Our semantics of a probabilistic database as a probability distribution over a set of deterministic databases is based on the possible worlds semantics [13] where subjective information, also called degrees of belief, is interpreted as a constraint over the probability distribution; we add the critical constraint on the expected cardinalities. Bacchus et al. [3] use the principle of entropy maximization to generate probability distributions from statistical knowledge. In their latter work [2], they consider the problem of generating probability distributions from subjective information using the principle of cross-entropy minimization. Again, this corresponds to our method of entropy maximization when a uniform prior distribution is assumed. Our Theorem 4.1 is an instance of Jeffrey's rule, described in [2].

There are various pieces of works that generate statistical/subjective information on databases. Many of the schema matching algorithms [18, 9, 20] return some score for the matched attributes, or even a probability [19]. A survey is in [10]. The recent CORDS system [14] detects correlations are soft functional dependencies between attributes.

7 Conclusions

We have shown that queries can be evaluated from statistics on the data, and from probabilistic views. We view this as an important component of a data integration system that copes with a variety of imprecisions: statistics and probabilities in views are among the hardest forms of imprecisions to use in query evaluation, and we have shown here how to model the problem, and how to solve it. Our work is foundational, and the general algorithms resulting from Corollary 3.2 and Theorem 4.1 run in exponential time. Yet we have shown some limited cases when efficient algorithms exists for computing queries from statistics and probabilistic views. We believe that other efficient methods will be discovered in the future.

Acknowledgment Frank Neven pointed us to the example in Sec. 1.2.

References

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison Wesley Publishing Co, 1995.

[2] Fahiem Bacchus, Adam Grove, Joseph Halpern, and Daphne Koller. Generating new beliefs from old. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 37–45, 1994.

[3] Fahiem Bacchus, Adam J. Grove, Joseph Y. Halpern, and Daphne Koller. From statistical knowledge bases to degrees of belief. *Artificial Intelligence*, 87(1-2):75–143, 1996.

[4] Daniel Barbará, Hector Garcia-Molina, and Daryl Porter. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng.*, 4(5):487–502, 1992.

[5] Roger Cavallo and Michael Pittarelli. The theory of probabilistic databases. In *VLDB'87, Proceedings of 13th Int. Conf. on Very Large Data Bases, September 1-4, 1987, Brighton, England*, pages 71–81, 1987.

[6] N. Dalvi, G. Miklau, and D. Suciu. Asymptotic conditional probabilities for conjunctive queries. In *ICDT*, 2005.

[7] N. Dalvi and D. Suciu. Probabilistic query answering using views. University of Washington Technical Report (TR 2005-06-03), 2005. http://www.cs.washington.edu/research/tr/techreports.html.

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

[9] AnHai Doan, Pedro Domingos, and Alon Y. Levy. Learning source description for data integration. In *WebDB (Informal Proceedings)*, pages 81–86, 2000.

[10] Philip A. Bernstein Erhard Rahm. A survey of approaches to automatic schema matching. *VLDBJ*, 10(4):334–350, 2001.

[11] Norbert Fuhr. Probabilistic datalog - a logic for powerful retrieval methods. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *SIGIR'95, Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA, July 9-13, 1995 (Special Issue of the SIGIR Forum)*, pages 282–290. ACM Press, 1995.

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

[13] Joseph Y. Halpern. An analysis of first-order logics of probability. In *IJCAI*, pages 1375–1381, Detroit, US, 1989.

[14] Ihab F. Ilyas, Volker Markl, Peter Haas, Paul Brown, and Ashraf Aboulnaga. Cords: automatic discovery of correlations and soft functional dependencies. In *SIGMOD*, pages 647–658, 2004.

[15] Laks V. S. Lakshmanan, Nicola Leone, Robert Ross, and V. S. Subrahmanian. Probview: a flexible probabilistic database system. *ACM Trans. Database Syst.*, 22(3):419–469, 1997.

[16] M. Lenzerini. Data integration: A theoretical perspective. In *PODS*, pages 233–246, 2002.

[17] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of the 22nd VLDB Conference, Bombay, India.*, 1996.

[18] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58, 2001.

[19] H. Nottelmann and N. Fuhr. The MIND architecture for heterogeneous multimedia federated digital libraries. In *Proceedings of Distributed Multimedia Information Retrieval*, pages 112–125, 2003.

[20] D. S. Luigi Palopoli and D. Ursino. Semi-automatic semantic discovery of properties from database schemas. In *IDEAS*, pages 244–253, 1998.

A Appendix

We sketch here the proof of Theorem 4.1. For each instance I, let $\Delta_I = \{i \mid V_i(I) = true\}$ be the unique set s.t. $V_\Delta(I)$ is true. The following follows from the Langrage multipliers (it generalizes Lemma 3.7):

Lemma A.1 *There exists a new set of statistics $\hat{\Sigma}$ and $m+1$ parameters f, and $C_1, \ldots, C_m$, such that the following holds. For every I s.t. $I \models \Gamma$:*

$$\mathbf{P}_{\Sigma,\Gamma,F}[I] \quad = \quad fC_\Delta \mathbf{P}_{\hat{\Sigma}}[I]$$

where $\Delta = \Delta_I$. For a query Q, it follows:

$$\mathbf{P}_{\Sigma,\Gamma,F}[Q] \quad = \quad \sum_\Delta fC_\Delta \mathbf{P}_{\hat{\Sigma}}[Q\bar{V}_\Delta \mid \Gamma] \qquad (13)$$

We abbreviate $\mathbf{P}_{\hat{\Sigma}}[- \mid \Gamma]$ with $\mathbf{P}[-]$. If the views are non-conflicting then $\mathbf{P}[Q\bar{V}_\Delta]$ in Eq.(13) is asymptotically equal to $\mathbf{P}[QV_\Delta]$. Substituting $Q \equiv true$ gives us an expression for f (since $\mathbf{P}[true] = 1$), and Eq.(13) becomes Eq.(14) below:

$$\mathbf{P}_{\Sigma,\Gamma,F}[Q] \quad = \quad \frac{\sum_\Delta C_\Delta \mathbf{P}[V_\Delta]\mathbf{P}[Q \mid V_\Delta]}{\sum_\Delta C_\Delta \mathbf{P}[V_\Delta]} \qquad (14)$$

Assume for the moment that all probabilistic facts are mutually independent, that is $\mathbf{P}_{\Sigma,\Gamma,F}[V_\Delta] = \prod_{j\in\Delta} \mathbf{P}_{\Sigma,\Gamma,F}[V_j] = p_\Delta$. We will prove later that this indeed holds. Substitute $Q = V_{\Delta_0}$ in (14), and note that $\mathbf{P}[V_{\Delta_0} \mid V_\Delta]$ is 1 when $\Delta_0 \subseteq \Delta$ and is asymptotically 0 otherwise (since $j \notin \Delta$ implies $\mu[V_j \mid V_\Delta] = 0$): this leads to (15) below, which, in turn, leads to (16) using an inclusion-exclusion argument:

$$\forall \Delta_0. \quad p_{\Delta_0} \quad = \quad \frac{\sum_{\Delta_0 \subseteq \Delta} C_\Delta \mathbf{P}[V_\Delta]}{\sum_\Delta C_\Delta \mathbf{P}[V_\Delta]} \qquad (15)$$

$$\frac{C_\Delta \mathbf{P}[V_\Delta]}{\sum_\Delta C_\Delta \mathbf{P}[V_\Delta]} \quad = \quad \sum_{\Delta \subseteq \Gamma} (-1)^{|\Gamma - \Delta|} p_\Gamma = \bar{p}_\Delta \qquad (16)$$

Substituting back in (14) and taking the limit $n \to \infty$ gives us a proof of Theorem 4.1. Once the formula Eq.(12) is derived, we can verify the independence assumption, asymptotically: computing $\mu_{\Sigma,\Gamma,F}[V_\Delta]$ with formula (12) gives us indeed p_Δ, since no view is probable given the others. The proof of the theorem and more details are in [7].