Probabilistic Query Answering Using Views

Nilesh Dalvi Dan Suciu*
University of Washington, USA.

Abstract

The paper studies two probabilistic query evaluation problems. The general setting is that we are given a probability distribution on all possible database instances and have to compute the probability of a tuple belonging to the query's answer. In the *deterministic view* problem, we are given a set of view instances and are asked to determine the probability of a tuple belonging to a query's answer in presence of data statistics and common world knowledge. This is related to the *open world assumption* in query answering using views. We show that the data complexity is NP-complete and identify important cases when it becomes PTIME, and when the query can be answered by a datalog program. In the second problem we consider, the views themselves are probabilistic: with uncertainties associated with the tuples in the views. It is unclear a priori what probability distribution on instances to assume here: we argue that a certain entropy-maximization distribution is the "right one", and show that the problem can be answered in this case (under some restrictions), albeit with a high complexity. However, in some cases that we identify, the problem can be reduced to the *deterministic view* problem and answered with a datalog program on a probabilistic database.

1 Introduction

A probabilistic database is a probability distribution on all instances [6, 4, 15, 12, 11, 8]. The early motivation for probabilistic databases was to model uncertainties at the tuple level: tuples are not known with certainty to belong to the database, or represent noisy measurements, etc. The type of probabilistic databases needed for such applications are tuple-independent probability distributions, which have a relatively simple semantics.

Our interest in probabilistic databases lies in the promise they hold in large scale data integration. Systems integrating dozens of databases, be it in the scientific domain or in a large corporation, need to cope with a wide variety of uncertainties. Examples of types of uncertainties include: different representations of the same object in different sources; imperfect and noisy schema

*Contact author: suciu@cs.washington.edu, tel. +1-206-685-1934, fax +1206-543-2969.

alignments; contradictory information across sources; constraint violations; 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. The probabilistic approach holds the promise of coping with all types of uncertainties in a uniform and principled way. Some recent projects are proposing to use the probabilistic method to large scale data integration systems [21, 5].

However, in this scenario the probability applies to the instance level, not to the tuple level. In particular it needs to allow correlations between tuples, both positive and negative. This leads to a more complex semantics than the simple, tuple-independent probabilistic databases studied in the past in the literature. This paper proposes a probabilistic model for large scale data integration, and studies the associated Probabilistic Query Answering Problem (PQAP). The model extends the Local As View (LAV) data integration paradigm [17, 16], by adding probabilities and statistics. To describe our approach, we briefly review LAV next.

Assuming m local data sources to be integrated, the LAV approach starts by defining a global mediated schema $\bar{R}$. Then it defines 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. To answer the query, the following definition is adopted. A global instance I is called *possible* if $J_i \subseteq v_i(I)$ for $i = 1, \ldots, m$. Given a query $q(\bar{R})$ on the global schema, a tuple t is called a *certain answer* if $t \in q(I)$ for all possible instances I. In LAV, the problem is to compute all certain answers for q from $\bar{J}$.

Our new probabilistic model can be described as an extension of LAV, in two ways: the local sources J_i are probabilistic, and there are statistics Σ on the global schema $\bar{R}$. The probability distribution on the global instance I is specified indirectly, through J_i, the view definitions, and the statistics Σ: as a result, complex correlations between tuples in I can be modeled this way. We call any probability distribution on global instances I *consistent* if it agrees with the probabilities observed in J_i, and with the statistics Σ. Of the many consistent distributions we pick the distribution $\mathbf{P}$ that maximizes the entropy: this is justified by the "principle of indifference", stating that, all things being equal, two unrelated events should have equal probabilities. The paper studies the Probabilistic Query Answering Problem: given q, compute $q(\bar{R})$ from the probabilistic local sources $\bar{J}$ and the statistics Σ.

The probabilistic model represents a new query answering paradigm, with roots in random graphs, 0/1-laws, and knowledge representation. This paper defines the model, and shows that query answering is decidable, and sometimes tractable.

1.1 An Example

Suppose we integrate two sources: $S_1(N, D)$ contains employee names N and departments D, and $S_2(D, B)$ contains departments D and buildings B. There are no constraints: an employee may work in several departments and a depart-

S_1	N	D	**P**
t_1	Larry Big	SalesDept	0.45
t'_1	Frank Little	HR	0.60

S_2	D	B	**P**
t_2	SalesDept	`EE1`	0.25
t'_2	HR	`EE1`	0.15
t''_2	HR	`MGH`	0.20

Figure 1: A view instance

ment may be spread across multiple buildings. We specify one global mediated relation $R(N, D, B)$, which lists employee names, their department, and their building. The mappings are given by the following views v_1, v_2:

$$S_1 : v_1(N_1, D_1) \quad :- \quad R(N_1, D_1, -)$$
$$S_2 : v_2(D_2, B_2) \quad :- \quad R(-, D_2, B_2)$$

A user wants to find all employees in building `EE1`. This is expressed as:

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

Suppose the views contain the tuples as shown in Fig. 1.1 and perhaps others (we make the Open World Assumption). We call this a *view instance*, and denote it with J. The sources are probabilistic: each tuple has an associated probability, shown above. Some possible reasons for these probabilities are: tuples may be the results of fuzzy matches between objects in yet other data sources; some values may be misspelled; the views may have been computed by queries with uncertain predicates. We are not concerned here with the source of these probabilities, but expect the system to take them into account when computing the probabilities of the query's answer.

Ignoring first the probabilities, we note that standard query answering semantics does not help us answer the query: neither `Larry Big` nor `Frank Little` is a *certain answer* to q, hence the system will return an empty set of answers. In our approach statistics on the mediated schema can be used to answer the query.

Using Statistics Suppose the system knows the following two statistics, which we denote collectively as Σ: the expected number of tuples in R is approximatively $\sigma = 800$; and the expected number of occurrences of each department in R is $\sigma_2 = 5$ times. Then `Larry Big` is an answer with probability 1/5. To see this imagine the five tuples in R that have $R.D =$ `SalesDept`. One has $R.N =$ `Larry Big`, and one has $R.B =$ `EE1`, and the probability that they are the same is 1/5. The same informal reasoning applies to `Frank Little`, and its probability is also 1/5. The system returns both answers, indicating their probabilities. In this simple example the two probabilities are equal, but in general they may differ, and the system ranks the answers according to their probabilities.

Our informal argument exposes two important assumptions that we will make throughout the paper. The first is that we need to know the statistics σ and σ_2 on the data. The role of σ_2 is clear, since the query's probability is $1/\sigma_2$. The role of σ is more subtle, since σ does not contribute directly to the query's probability. However, if we knew nothing about the expected size of R, then, by the principle of indifference, *every* tuple over the domain of data values belongs to R with probability $1/2$. Then, every name in the domain (not just Larry Big or Frank Little, but any name, say, from the phone book) has a probability $1 - 1/2^n \approx 1$ of being an answer, where n is the number of all possible department names in the domain[1]. Without assuming a reasonable bound on the size of R we cannot answer the query in practice. Thus, in our theoretical development we will always assume σ (the expected cardinality of a table) to be given, directly or indirectly, and that it is much smaller than n (the size of the domain)[2]. Often, however, the probabilities we compute do not depend on the exact value of some of the statistics.

The second important assumption is that the domain of all data values is very large: this is just the other half of the statement above, saying that R is small compared to the domain. We will assume that all attributes in all relations have the same[3] domain D, and that $n \to \infty$, where $n = |D|$. As a consequence, if we don't know anything about a database instance I, then a tuple t has probability ≈ 0 of belonging to I. This is an important property, distinguishing probabilistic databases from knowledge representation [3] where, for example, if x is a randomly chosen value from the universe, the probability of $Male(x)$ is $\approx 1/2$.

Using Explicit Probabilities The meaning of the probabilities listed in the instance J is the following. There exists some probability distribution on all possible instances of R such that, after computing the views v_1 and v_2 the five tuples in J have exactly the probabilities listed in the figure. This imposes some strict constraints on the probability distribution on R, in addition to those imposed by the statistics Σ. For example, the sum of the probabilities of all instances that contain some tuple $(\texttt{Larry Big}, \texttt{SalesDept}, -)$ is exactly 0.45, etc. Clearly, this distribution is not tuple-independent, because there is interdependence between the tuples mentioned in J. The statistics requirements add further correlations between tuples. The system needs to compute the answers to q and their probabilities from the statistics and the view instance J and their probabilities.

[1] For any person name p and department d the probability of $(p, d, \texttt{EE1}) \in R$ is $1/2$, implying that the probability of $\exists d.(p, d, \texttt{EE1}) \in R$ is $1 - 1/2^n$.

[2] On other hand, σ_2 is not critical. If missing, the expected number of occurrences of each department is 1, and the probabilities of both answers increase to ≈ 1. We have studied this case in [7].

[3] Typed attributes with multiple domains can also be handled (if all are big), but we restrict to a single domain for presentation purposes.

2 Problem Definition (PQAP)

We define here our probabilistic model for data sharing, and define the query answering problem. The model borrows ideas from both probabilistic databases [12] and from models of belief [3], and adds specific features motivated by global data sharing.

2.1 Preliminary Definitions

Basic Notations D denotes the finite domain, and its cardinality is $n = |D|$. $R_1, \ldots, R_k$ are the relation names in the relational schema. $Attr(R_i)$ denotes the set of attributes of R_i. Tuples are written in datalog notation, e.g. $R_1(a, b, c), R_2(b, b), R_3(a, b, c)$, and $Tup(R_i)$ is the set of all tuples over relation R_i, while $Tup = \bigcup_{i=1,k} Tup(R_i)$ is the set of all tuples. There are $|Tup| = \sum_i n^{arity(R_i)}$ possible tuples over the domain D. A data instance I is a set of tuples, $I \subseteq Tup$ and $R_1^I, \ldots, R_m^I$ denotes the relational instances for I. We write $Inst$ $(= \mathcal{P}(Tup))$ for the set of all instances. We will consider only conjunctive queries throughout the paper, unless otherwise stated, and denote them as single-rule datalog programs. Variables are letters from the end of the alphabet $(x, y, z, u, v, \ldots)$, constants are letters from the beginning of the alphabet $(a, b, c \ldots)$, and we use standard terminology such as head variables, body, subgoals, etc. All our results in this paper carry over to conjunctive queries with the $\neq$ predicate, but for simplicity we omit discussing such queries here. When a query is boolean we will denote it with an upper case letter.

An important transformation that we will do repeatedly in the paper is to convert view instances into a boolean query. Specifically, given a view v with head variables $\bar{x}$, we want to say that the tuple t belongs to its answer: we can express that with the boolean conjunctive query $v[t/\bar{x}]$. If $J = \{t_1, \ldots, t_m\}$ is a set of tuples, we want to say that J is a subset of v's answer. This is standard in the Open World Assumption. We can say this either through m boolean queries $v[t_i/\bar{x}]$, $i = 1, m$, or with a single big boolean query given by their conjunction. To illustrate, consider the view: $v(x, y) \leftarrow R(x, a, z), S(z, y)$ and $J = \{(a, b), (c, b)\}$, then the boolean query is given by: $V \leftarrow R(a, a, z_1)$, $S(z_1, b)$, $R(c, a, z_2)$, $S(z_2, b)$.

Probabilistic Databases We define next:

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

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

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:

$$\mathbf{P}[P_0|P] = \frac{\mathbf{P}[P_0P]}{\mathbf{P}[P]}$$

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

where $P_0P = P_0 \wedge P$ 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.

Constraints We restrict the constraints to functional dependencies (FD), which we denote as usual $\bar{A} \rightarrow \bar{B}$, where $\bar{A}$ and $\bar{B}$ are sets of attributes. Denote Γ a (possibly empty) set of FDs. We write $I \models \Gamma$ whenever the instance I satisfies Γ, and write $\mathbf{P} \models \Gamma$ whenever the probabilistic database $\mathbf{P}$ has the property: $\forall I.\mathbf{P}(I) > 0 \Rightarrow I \models \Gamma$ (i.e. $\mathbf{P}[\Gamma] = 1$).

Statistics We consider two kinds of statistics in this paper, cardinalities and fan-outs. A *cardinality statistics* on a relation R is a statement of the form $card_R(\bar{B}) = \sigma$, where $\bar{B} \subseteq Attr(R)$ and $\sigma > 0$ is a number. A probability distribution $\mathbf{P}$ is consistent with this statistics if $E[|\Pi_{\bar{B}}(R^I)|] = \sigma$. When $\bar{B} = Attr(R)$ then the statistics simply asserts the size of R and we write it $card(R) = \sigma$.

A *fanout statistics* is a statement of the form $fanout_R(\bar{A} \Rightarrow \bar{B}) = \sigma$, where $\bar{A}, \bar{B} \subseteq Attr(R)$ and $\sigma > 1$ is a number. We define its meaning next. Given an instance I and an A-tuple $\bar{a}$, the fanout of R^I at $\bar{a}$ is:

$$fanout_{R,\bar{a}}[\bar{A} \Rightarrow \bar{B}](I) = |\Pi_{\bar{A},\bar{B}}(\sigma_{\bar{A}=\bar{a}}(R^I))|$$

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:

$$\forall \bar{a}. \quad E[fanout_{R,\bar{a}}[\bar{A} \Rightarrow \bar{B}] \mid \bar{a} \in \Pi_{\bar{A}}(R^I)] = \sigma \quad (4)$$

We denote Σ a set of statistics for our relational schema and write $\mathbf{P} \models \Sigma$ whenever $\mathbf{P}$ satisfies all statistics in Σ. We have argued in Sec. 1.1 for the need of an upper bound on the expected size of each relation. For that, we will require that Σ "covers" all attributes in all relations. However, some sets of fanout statistics are quite hard to analyze: for example, given $R(A, B)$ and $card_R(A) = \sigma_1$, $card_R(B) = \sigma_2$, it seems difficult to compute the expected size of R. For that reason we will make the following simplifying assumption throughout the paper: that for each relation R the fanout statistics on R form

chain, covering all attributes in R exactly once. More precisely, we require Σ to contain precisely the following statistics about R, and no others:

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

where $k \geq 1$, and $\bar{A}_1 \cup \ldots \cup \bar{A}_k$ is a partition of $Attr(R)$. With this restriction it is always possible to compute the expected size of R:

Proposition 2.2. *If* $\mathbf{P}$ *is any distribution consistent with the above statistics on* R*, then the expected size of* R *is* $\prod_{i=1}^{k} \sigma_i$*.*

Proof. Let $\bar{B}_i = \bigcup_{j \leq i} \bar{A}_j$ and let $card_R[B_i]$ denote the number of distinct values of attributes B_i. By definition, $E[card_R(\bar{B}_1)] = E[card_R(\bar{A}_1)] = \sigma_1$.

Next, we show that for $i \geq 1$, $E[card_R(\bar{B}_{i+1})] = \sigma_i * E[card_R(\bar{B}_i)]$, which establishes the proposition.

If $\bar{b_i} = (\bar{a_1}, \bar{a_2}, \cdots, \bar{a_i})$ is a vector of values for attributes $\bar{B}_i$, let $P[\bar{B}_i = \bar{b_i}]$ denote the probability that $\bar{b_i}$ occurs among the distinct values of $\bar{B}_i$. By linearity of expectations, we have

$$\begin{aligned} E[card_{i+1}] &= \sum_{\bar{b}_{i+1}} P[\bar{B}_{i+1} = \bar{b}_{i+1}] \\ &= \sum_{\bar{b}_{i+1}} P[\bar{B}_i = \bar{b}_i] * P[\bar{A}_{i+1} = \bar{A}_i | \bar{B}_i = \bar{b}_i] \end{aligned}$$

Here $\bar{b}_{i+1}$ range over all possible values of $\bar{B}_{i+1}$ from the domain. Grouping the terms by distinct values of $\bar{B}_i$, we have

$$\begin{aligned} E[card_R(\bar{B}_{i+1})] &= \sum_{\bar{b}_i} \left(P[\bar{B}_i = \bar{b}_i] * \sum_{\bar{a}_{i+1}} P[\bar{A}_{i+1} = \bar{A}_i | \bar{B}_i = \bar{b}_i] \right) \\ &= \sum_{\bar{b}_i} P[\bar{B}_i = \bar{b}_i] * \sigma_{i+1} \\ &= \sigma_{i+1} * \sum_{\bar{b}_i} P[\bar{B}_i = \bar{b}_i] \\ &= \sigma_{i+1} * E[card_R(\bar{B}_i)] \end{aligned}$$

The theorem then follows from the fact that the expected size of R is just $E[card_R(\bar{B}_k)]$. □

As a simple example, the following two statistics model the scenario in our motivating example.

$$\begin{aligned} card_R[D] = \sigma_1 = \sigma/\sigma_2 &= 800/5 = 160 \\ fanout_R[D \Rightarrow N, B] = \sigma_2 &= 5 \end{aligned}$$

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

Finally, we note that a functional dependency and a fanout statistics may conflict. For example no probability distribution is consistent with both $A \to B$ and $fanout(A \Rightarrow B) = 2$. To eliminate such cases, we assume that whenever we have a functional dependency where A occurs on the left and B on the right, then the stratum of A is higher than or equal to that of B: i.e. $A \in \bar{A}_i$, $B \in \bar{A}_j$ and $i \geq j$.

Probabilistic Facts Given a view v, a tuple t, and a probability $p \in [0, 1]$, a *probabilistic fact*, or *probabilistic view* is a statement of the form $\mathbf{P}[t \in v] = p$. We have discussed how to convert the statement $t \in v$ into a conjunctive query $V = v[t/\bar{x}]$. Hence, a probabilistic fact can also be expressed as $\mathbf{P}[V] = p$. It is thus an assertion on the probability distribution that states that the sum of probabilities of all data instances satisfying V is p. When $p = 1$ then we call it a *deterministic fact* (view). We will 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 in $\mathbf{P}$. It is important to keep in mind that F is derived from two separate entities: a set of view definitions $v_1, v_2, \ldots$, and a view instance J, which is a set of tuples with associated probabilities. When we study the data complexity we keep the views fixed and vary J.

The Problem We will now state formally the Probabilistic Query Answering Problem studied in this paper, denoted PQAP. We are given a triple (Γ, Σ, F), and a boolean query Q. Consider all probabilistic databases $\mathbf{P}$ that satisfy:

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

and denote $\mathbf{P}_{\Gamma,\Sigma,F}$ the distribution that has the maximum entropy H. The problem is to 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 the probability is $\mu_{\Gamma,\Sigma,F}[t \in v] > 0$.

Justification Maximizing the entropy is justified by the *principle of indifference*, which states that we should assume equal probabilities if there is no evidence to the contrary. We argue that it is a reasonable definition to adopt. We have done some preliminary inquiries into alternative definitions, e.g. starting from some canonical distribution then seeking to minimize the cross-entropy, and have found that it leads only to slight variations in $\mu[Q]$.

We take the limit $n \to \infty$ to study the behavior when the domain of databases is very large compared to active domain, as we discussed in Sec 1.1. Also, $\mathbf{P}_{\Gamma,\Sigma,F}(Q)$ is a very complex expression in n, and it is impossible to compute it exactly in most practical cases. The limit expression, however, is computable, as we will show.

Finally, we will analyze the complexity of the PQAP in two ways. The *query complexity* is the complexity measured as a function of the sizes of Γ, Σ, F, and Q. The *data complexity* is the complexity measured as a function of the size of J, the view instances that were used to build the probabilistic facts F.

Outline In the rest of the paper we study/solve several cases of the PQAP. These cases can be classified into two groups:
Deterministic Views All facts in F are deterministic. Here we show (1) there exists an algorithm for computing $\mu_{\Gamma,\Sigma,F}[t \in q]$, (2) under certain restrictions, there exists a rewriting of q in datalog over the instance J that returns all tuples t that are *probable* ($\mu[t \in q] > 0$) or *almost certain* ($= 1$). (3) in general, both query and data complexity are NP-hard,
Probabilistic Views Here we show (1) there exists an algorithm that computes $\mu_{\Gamma,\Sigma,F}[t \in q]$. (2) the query above for the almost certain answers, when evaluated on the probabilistic instance J, returns the set $(t, \mu_{\Gamma,\Sigma,F}[t \in q])$.

3 Deterministic Views

We consider here the case when there are only deterministic views (deterministic facts). We identify the set of deterministic facts with one boolean conjunctive query V, e.g. by converting a set of statements of the form $t_i \in v_j$. Recall that Γ is a set of functional dependencies, Σ is a set of statistics, and n is the size of the domain D. Our general problem is to compute $\mu_{\Gamma,\Sigma,V}[Q] = \lim_{n\to\infty} \mathbf{P}_{\Gamma,\Sigma,V}[Q]$. We call a tuple t a *probable answer* to q if $\mu_{\Gamma,\Sigma,V}[t \in q] > 0$; we call it an *almost certain* answer if $\mu_{\Gamma,\Sigma,V}[t \in q] = 1$. When V is derived from a view instance J, we will consider the problem of computing all probable answers, or all almost certain answers from J.

We proceed in three steps. First we study the probabilities of queries under a specific binomial distribution based on Σ. Next, we show the relation of this distribution to the entropy-maximizing distribution and describe how to compute $\mathbf{P}_{\Gamma,\Sigma,V}$. Third, we study this as a function of J.

3.1 The Binomial Distribution

The *binomial distribution* $\mathbf{P}$ introduced here is associated to a set of statistics Σ. We will define it, then show how to compute $\lim_{n\to\infty} \mathbf{P}[Q \mid V, \Gamma]$ for conjunctive queries Q, V

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, $card_R[A_1, \ldots, A_m] = \sigma$. The binomial distribution is the following. For each tuple in D^m set its probability $p = \sigma/n^m$ (assuming n "large enough", i.e. $n > \sigma^{1/m}$). 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 $1-(1-\sigma/n^m)^{n^m}$; the expected cardinality of a nonempty I is $\sigma/(1-(1-\sigma/n^m)^{n^m})$. Consider the function $f(x) = x/(1-(1-x/n^m)^{n^m})$, $x \geq 0$. It is strictly increasing, and $f(0) = 1$, hence $f(x) = \sigma$ has a unique solution $\forall \sigma > 1$, denote it $\hat{\sigma}$. Consider the binomial distribution $\hat{\sigma}$: the expected size of a nonempty I is σ.

Consider now an arbitrary set of statistics on R:

$$\begin{aligned} card_R(\bar{A}_1) &= \sigma_1 \\ fanout_R(\bar{B}_{i-1} \Rightarrow \bar{A}_i) &= \sigma_i \ (2 \leq i \leq k) \end{aligned}$$

where $\bar{B}_i = \bigcup_{j \leq i} A_j$. We define the following distribution, which we still call "binomial". Denote $m_i = |\bar{B}_i|$, 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 $\mathbf{P}$, which satisfies all the fanout constraints. For any instance I, there is an explicit formula for $\mathbf{P}[I]$, which we give now. Let $N_i = n^{|\bar{A}_i|}$ denote the size of the universe for attributes $\bar{A}_i$. Let $s_i = |I^{(i)}|$. Also, let $p_1 = \sigma_1/N_1$ and $p_i = \hat{\sigma}_i/N_i$ for $i \geq 2$. Then, we have

$$\mathbf{P}[R = I] = \prod_{i=1,k} \frac{p_i^{s_i}(1-p_i)^{N_i - s_i}}{(1-(1-p_{i+1})^{N_{i+1}})^{s_i}} \tag{5}$$

In the above equation, the term in the numerator is the probability under binomial distribution for occurance of the values of $\bar{A}_i$. The term in the denominator accounts for the fact that for each value of $\bar{A}_i$, we are choosing a non-empty set of values for attributes $\bar{A}_{i+1}$. Similarly, we have the following

$$\mathbf{P}[R \supseteq I] = \prod_{i=1,k} \left(\frac{p_i}{1-(1-p_{i+1})^{N_{i+1}}}\right)^{s_i} \tag{6}$$

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 Query Probability

We show here how to compute $\mathbf{P}[Q]$ and $\mu[Q \mid V]$ for a binomial distribution. The explicit formulas we give are very powerful, since these probabilities are almost impossible to compute with brute force. Importantly, we need these formulas later, when we study query answering over probabilistic facts. The results here are non-trivial extensions of earlier results in [7].

We first give a overview of our plan to compute $\mathbf{P}[Q]$. For every boolean conjunctive query Q, we define two parameters, $A(Q)$ and $C(Q)$, which we call the *arity* and *coefficient* of Q. The parameters are defined in such a way that

$\mathbf{P}[Q]$ is closely related to the quantity $C(Q)n^{A(Q)}$ and an exact value is obtained by summing this quantity over all *unifications* of Q.

We now formally state the method to compute $\mathbf{P}[Q]$. We have a set of functional dependencies Γ and a set of statistics Σ. We need to introduce several notations. Consider one relation name R in the relational schema, and assume that the statistics in Σ define the following partition on its attributes: $\bar{A}_1, \ldots, \bar{A}_k$. Define $\bar{B}_i = \bigcup_{j \leq i} \bar{A}_j$. 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. Define the *proper arity* of $R^{(i)}$ to be $A(R^{(i)}) = |\bar{A}_i|$. Also, define the *proper attributes* of $R^{(i)}$ to be the attributes $\bar{A}_i$.

If Q is a boolean conjunctive query Q, we view it as a canonical database, by interpreting each variable in its body as a constant: hence R^Q denotes all subgoals in Q referring to the relation name R. Denote $Q^{(*)}$ a new query, over the extended relational schema, whose canonical database is the following: for each relation R, the instance of $R^{(i)}$ is $\Pi_{\bar{B}_i}(R^Q)$, for $i = 1, \ldots, k$. Notice that $Q^{(*)}$ has the same variables and constants as Q, and that it contains all subgoals of Q (since $R^{(k)} = R$), plus new subgoals referring to the new relation names. By abuse of notation, if g is a subgoal in some query, we denote $A(g)$ the proper arity of the relation to which g refers. We define the *arity* of Q to be:

$$A(Q) \quad = \quad \sum_{g \in subgoals(Q^{(*)})} A(g) \tag{7}$$

Given a query Q, we call a variable *free* if it occurs among the proper attributes of only a single sub-goal of $Q^{(*)}$. We call a sub-goal of $Q^{(*)}$ a *trivial subgoal* if all of its proper attributes have free variables.

We now define the coefficient of a query Q. To start with, we assume that $Q^{(*)}$ does not contain any trivial sub-goals as they need different treatment. Let $\sigma_1, \ldots, \sigma_k$ be the numbers used in the k statistics on R. For each i, we define the *coefficient* of $R^{(i)}$ to be $C(R^{(i)}) = \sigma_i/(1 - e^{-\sigma_{i+1}})$.[4] Again by abuse of notation, if g is a subgoal in some query, we denote $C(g)$ the coefficient of the relation to which g refers. If $Q^{(*)}$ does not contain trivial sub-goals, we define the *coefficient* of Q to be:

$$C(Q) \quad = \quad \prod_{g \in subgoals(Q^{(*)})} C(g) \tag{8}$$

We have the following lemma.

Lemma 3.1. *Let Q be any boolean conjunctive query containing only ground terms, i.e. no variable symbols. Then, $\mu_n(Q)$ is asymptotically equal to $C(Q)/n^{A(Q)}$. Note that Q does not contain any trivial subgoals.*

Proof. Let I denote the canonical database for Q. Apply Eq 6 and the definitions of $C(Q)$ and $A(Q)$. □

[4]For an intuition, look at Eq (6), describing the probability of I viewed as a query, where the numerator contributes σ_i to the coefficient and the denominator contributes to $(1 - e^{-\sigma_{i+1}})$.

We now describe how to handle trivial sub-goals. Consider all trivial subgoals corresponding to the relation $R^{(i)}$ and let them be l in number. They together contribute the following single term in the product for $C(Q)$:

$$1 - e^{-\sigma_i} \sum_{0 \leq j \leq l} \frac{(\sigma_i)^j}{j!} \tag{9}$$

Thus, $C(Q)$ is the product of $C(g)$ for each non-trivial subgoal g in $Q^{(*)}$ and the terms in Eq 9 for each corresponding group of trivial sub-goals.

Define a *substitution* h on a query Q to be any function from the variables in Q to variables and constants in Q. Also, define $h(Q)$ to be the query obtained by applying the substitution to each variable in Q: $h(Q)$ partitions the subgoals of Q into equivalence classes, where g and g' are in the same equivalence class if $h(g) = h(g')$. Call h 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 call a query G a *most general unifying query* for Q if $G = h(Q)$ where h is some most general unifier 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. Suppose now that we have one 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 denote $MGUQ_\Gamma(Q)$ the set of most general unifying queries of Q (we include only one copy up to isomorphism) that satisfy Γ (when viewed as canonical databases). Its size is at most exponential in Q.

For a query $G \in MGUQ_\Gamma(Q)$ let $V(G)$ denote the number of distinct variables in G, and $D(G) = A(G) - V(G)$ the *degree* of G. For a query Q define:

$$\begin{aligned} exp_\Gamma(Q) &= \min\{D(G) \mid G \in MGUQ_\Gamma(Q)\} \\ MGUQ^0_\Gamma(Q) &= \{G \mid G \in MGUQ_\Gamma(Q), D(G) = exp_\Gamma(Q)\} \\ coeff_\Gamma(Q) &= \sum\{C(G) \mid G \in MGUQ^0_\Gamma(Q)\} \end{aligned}$$

The following gives the query probability:

Theorem 3.2. *Let Σ be a set of statistics, $\mathbf{P}$ the binomial distribution for Σ, and Γ a set of functional dependencies. Let Q be a conjunctive query. Then:*

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

The proof is deferred to Section 5.1. In the sequel we will drop the subscript Γ when $\Gamma = \emptyset$, e.g. write $MGUQ(Q)$.

Example 3.3 Let $R(A, B)$ be a binary relation, assume no functional dependencies and one single cardinality constraint $card_R(A, B) = \sigma$. Consider the query:

$$Q = R(a, -), R(-, b)$$

We make the anonymous variables explicit, denoting the query $R(a,y), R(x,b)$. Then $MGUQ(Q)$ contains exactly two queries:

$$\begin{aligned} Q &= R(a,y), R(x,b) \\ Q' &= R(a,b) \end{aligned}$$

Both have $D(Q) = D(Q') = 2$ (since the true arity of R is 2, and Q has 2 variables, Q' has none), hence $MGUQ^0(Q) = MGUQ(Q)$. The coefficients are $C(Q) = \sigma^2$, $C(Q') = \sigma$. Hence $exp(Q) = 2$,$coeff(Q) = \sigma + \sigma^2$, and by Theorem3.2:

$$\mathbf{P}[Q] = \frac{\sigma^2 + \sigma}{n^2} + O(\frac{1}{n^3})$$

To appreciate the power of this theorem, let us compute $\mathbf{P}[Q]$ directly, using the fact that each tuple has an independent probability $p = \sigma/n^2$. We get the following exact formula:

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

which can be simplified to $(\sigma^2 + \sigma)/n^2 + O(1/n^3)$. This brute force approach becomes intractable for more complex queries, or more complex statistics.

It follows immediately how to compute the limit conditional probability $\mu[Q|V,\Gamma] = \lim_{n\to\infty} \mathbf{P}[Q|V,\Gamma]$:

Corollary 3.4.

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

Example 3.5 Consider our example in Sec. 1.1: we have a ternary relation $R(N,D,B)$ and the two statistics: $card_R(D) = \sigma_1$, $fanout_R(D \Rightarrow N, B) = \sigma_2$. Consider the views v_1, v_2 defined in Sec. 1.1, and assume we know only two deterministic facts: $t_1 \in v_1$ and $t_2 \in v_2$. We want to compute the probability that `LarryBig` is in `EE1`. This corresponds to the following boolean view V and boolean query Q:

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

We start by computing $V^{(*)}$:

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

The arity of V is $A(V) = 1 + 2 + 2 = 5$, hence its degree is $D(V) = 5 - 2 = 3$, since V has two variables; its coefficient is $C(V) = \sigma_1\sigma_2^2$. $MGUQ(V)$ contains

two queries, namely V itself and W :- $R(\texttt{LarryBig}, \texttt{SalesDept}, \texttt{EE1})$, and both have $D(V) = D(W) = 3$. Hence:

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

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:

$$\begin{aligned} \mathit{exp}(QV) &= 3 \\ \mathit{coeff}(QV) &= \sigma_1\sigma_2 \end{aligned}$$

It follows that $\mu[Q|V] = 1/(1+\sigma_2)$. This is a precise answer that is slightly different from the informal result $1/\sigma_2$ obtained in Sec. 1.1, as there we did not account for the very small probability that the `SalesDept` may have many more than σ_2 employees: this contributes to a decrease from $1/\sigma_2$ to $1/(1+\sigma_2)$.

Example 3.6 Functional dependencies affect μ, as the following example illustrates. Assume $R(A, B, C, D, E)$ with cardinality statistics $\mathit{card}(R) = \sigma$, and consider the view and query:

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

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

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

$D(V_1) = D(V_2) = \mathit{exp}(V) = 15$, 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 $\mathit{exp}(-)$.

Definition 3.7. *Let Q, V be two queries.*

- *Q is called* probable *given V if $\mu[Q \mid V, \Gamma] > 0$.*
- *Q is called* almost certain *given V if $\mu[Q \mid V, \Gamma] = 1$.*

Consider a non-boolean view q; a tuple t is called:

- *a* probable answer*, if $\mu[t \in q \mid V, \Gamma] > 0$.*
- *an* almost certain answer*, if $\mu[t \in q \mid V, \Gamma] = 1$.*

For an illustration, assume a cardinality statistics $card(R) = \sigma$. Consider first $V : -R(a,-), R(-,b)$ and the query $q(x,y) :\text{-}\ R(x,y)$. Then the tuple $t = (a,b)$ is a probable answer to q, since $\mu[t \in q \mid V] = 1/(1+\sigma)$. Consider now $V :\text{-}\ R(a,b,-), R(-,b,c)$ and the query $q(x,z) :\text{-}\ R(x,-,z)$. Then (a,c) is an almost certain answer to q, since $\mu[t \in q] = \sigma/\sigma = 1$: note that it is not a *certain answer* [1].

The following is a characterization of these two properties:

Proposition 3.8. *For any conjunctive queries Q, V:*

1. *The following statements are equivalent:*
   (a) $\mu[Q \mid V] > 0$
   (b) $MGUQ^0_\Gamma(QV) \subseteq MGUQ^0_\Gamma(V)$
   (c) $\exists U \in MGUQ^0_\Gamma(V)$ *s.t. there exists a homomorphism* $h : Q \to U$.
2. $\mu[Q \mid V] = 1$ *iff* $MGUQ^0_\Gamma(QV) = MGUQ^0_\Gamma(V)$.

Proof.

1. We will prove this by showing $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$.

   $(a) \Rightarrow (b)$ Lets assume that $\mu[Q \mid V] > 0$. Then, we have $exp_\Gamma(QV) = exp_\Gamma(V)$. Let $\eta(QV)$ be any query in $MGUQ^0_\Gamma(QV)$. Thus, $D(\eta(QV)) = exp_\Gamma(QV) = exp_\Gamma(V)$. Since $goals(\eta(V)) \subseteq goals(\eta(QV))$, by Lemma 5.6, $D(\eta(QV)) \geq D(\eta(V))$. Thus, $exp_\Gamma(V) \geq D(\eta(V))$, which shows that they are equal (since $exp_\Gamma(V) \leq D(\eta(V))$ for all η). Thus, $D(\eta(QV)) = D(\eta(V)$ and again by Lemma 5.6, $\eta(QV)$ and $\eta(V)$ are equivalent. Also, since $D(\eta(V)) = exp_\Gamma(\eta(V))$, $\eta(V) \in MGUQ^0_\Gamma(V)$. Thus, we have shown that for every query in $MGUQ^0_\Gamma(QV)$, there is an equivalent query in $MGUQ^0_\Gamma(V)$. Hence, $MGUQ^0_\Gamma(QV) \subseteq MGUQ^0_\Gamma(QV)$.

   $(b) \Rightarrow (c)$ Since $MGUQ^0_\Gamma(QV) \subseteq MGUQ^0_\Gamma(QV)$, there exists a query $\eta(QV)$ in $MGUQ^0_\Gamma(QV)$ that also belongs to $MGUQ^0_\Gamma(V)$. The above argument shows that $\eta(QV)$ is equivalent to $\eta(V)$ and $\eta(V) \in MGUQ^0_\Gamma(V)$. If we put $U = \eta(V)$ and $h = \eta$, we see that h has to be a homomorphism from Q to U.

   $(c) \Rightarrow (a)$ Let $U = \eta(V)$. Thus, h is a homomorphism from Q to $\eta(V)$. Consider the query $h(\eta(VQ))$. It is equivalent to U. Hence, $exp_\Gamma(QV) \leq D(U) = exp_\Gamma(V)$, which means $\mu[Q \mid V] > 0$.

2. For $\Rightarrow$ direction, suppose $\mu[Q \mid V] = 1$. By part 1, we have $MGUQ^0_\Gamma(QV) \subseteq MGUQ^0_\Gamma(QV)$. Also, $coeff_\Gamma(QV) = \sum\{C(G) \mid G \in MGUQ^0_\Gamma(QV)\}$ and $coeff_\Gamma(V) = \sum\{C(G) \mid G \in MGUQ^0_\Gamma(V)\}$. For them to be equal, we must have $MGUQ^0_\Gamma(QV) = MGUQ^0_\Gamma(QV)$. The other direction holds trivially.

□

3.2 The Entropy Maximization Distribution

We now return to our original goal in the PQAP, of computing the entropy maximization distribution: so far we have shown only how to compute the binomial distribution. We show here that they are approximatively equal. We start by showing that $\mathbf{P}$ and $\mathbf{P}_\Sigma$ (i.e. no FD's, no view) are equal.

Proposition 3.9. $\mathbf{P}_\Sigma = \mathbf{P}$

Proof. (Sketch) To give a flavor of the proof, we consider the case when there is a single relation $R(A_1, \ldots, A_m)$, and Σ contains only the following:

$$\begin{aligned} card_R(A_1) &= \sigma_1 \\ fanout_R(A_1 \Rightarrow A_2) &= \Sigma_1 \end{aligned}$$

For any value a of attribute A_1, let $\chi_{[A_i=a]}$ denote the function on the set of database instances that takes value 1 if I contains a tuple with $I.A_i = a$ and 0 otherwise.

Now, $\mathbf{P}_\Sigma$ is defined by:

$$f_1(\bar{x}) = \sum_I x_I - 1 = 0 \tag{10}$$

$$f_2(\bar{x}) = \sum_I x_I card_R(A_1) - \sigma_1 = 0 \tag{11}$$

$$\forall a : f_a(\bar{x}) = \sum_I x_I \chi_{[A_1=a]}(I) fanout_R(A_1 \Rightarrow A_2) - \sigma_2 = 0 \tag{12}$$

$$H(\bar{x}) = \sum (x_I \log 1/x_I) = \text{ is maximized} \tag{13}$$

By the Lagrange multipliers method, $\exists \lambda_1, \lambda_2$ and λ_a for each a such that:

$$\forall I. \quad \frac{\partial H}{\partial x_I} - \lambda_1 \frac{\partial f_1}{\partial x_I} - \lambda_2 \frac{\partial f_2}{\partial x_I} - \sum_a \lambda_a \frac{\partial f_a}{\partial x_I} = 0$$

By substituting the functions in the above equation and simplifying, we see that $\mathbf{P}_\Sigma$ is off the form:

$$x_I = AB^{card_R A_i} \prod_{a \in R.A_i} C^{fanout_R(A_1 \Rightarrow A_2)}$$

for some constants A,B and C. If we examine Eq (5) for $\mathbf{P}$, we see that function $\mathbf{P}$ has the same form as above. Also, we know that $\mathbf{P}$ satisfies Equations (10)-(13), since $\mathbf{P}$ by construction, satisfies all the statistics. Hence, $\mathbf{P} = \mathbf{P}_\Sigma$.

□

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 $\mathbf{P}_\Sigma[Q \mid V, \Gamma]$. Since both Γ and V are boolean properties on instances, the two steps are instances of the following lemma:

Lemma 3.10. *Let Σ be a set of statistics, and let P_1, P_2 be any two boolean properties on instances. Then there exists a different set of statistics $\hat{\Sigma}$ s.t. for any boolean query*[5] *Q:*

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

$\hat{\Sigma}$ is related to Σ in the following way. Let $S(I) = \sigma$ be in Σ, where $S(I) = card_R(I)$ or $fanout_{R,\bar{a}}(I)$. Then the expected value of S on all instances satisfying P_2 is σ, i.e.:

$$E_{\hat{\Sigma},P_1}[S|P_2] \quad = \quad \sigma \tag{14}$$

Proof. (Sketch) We illustrate the first part only for the case treated in the proof of Prop 3.9, and we assume $P_1 = true$. Denote $P = P_2$ and χ_P the characteristic function of P. We proceed as in the proof of Prop 3.9, adding a new constraint, $f_3(\bar{x}) = \sum_I \chi_P x_I - 1 = 0$. The solution x_I is now of the form $x_I = \chi_P A B^{card_R A_i} \prod_{a \in R.A_i} C^{fanout_R(A_1 \Rightarrow A_2)}$. Hence $\mathbf{P}_{\Sigma,P}$ looks like some binomial distribution (but with different statistics), only it drops to 0 on instances that don't satisfy P: $\mathbf{P}_{\Sigma,P}[I]$ is $f\mathbf{P}_{\hat{\Sigma}}[I]$ when $I \models P$ and 0 otherwise. Here f is a normalization factor, which follows immediately to be $1/\mathbf{P}_{\hat{\Sigma}}[P]$. Thus $\mathbf{P}_{\Sigma}[I] = \mathbf{P}_{\hat{\Sigma}}[I \mid P]$. The second statement is immediate. □

We consider now the relationship between Σ and $\hat{\Sigma}$. One can think of Σ as a perturbation of $\hat{\Sigma}$ (and not the other way around), given by Equation (14). It turns out that the perturbation is small, but the exact relationship is rather complex. 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. Recall that we assimilate every query G with its canonical database. Define:

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

Then, we prove the following in the full version of the paper:

Proposition 3.11. *For every $i = 1, \ldots, k$:*

$$\sigma_i \quad = \quad \hat{\sigma}_i + \frac{\sum_{G \in MGUQ^0_\Gamma(V)} G_i(G) C(G)}{\sum_{G \in MGUQ^0_\Gamma(V)} C(G)} \tag{15}$$

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

We defer the proof to Sec 5.1 (see Corollary 5.14).

To find $\hat{\Sigma}$ one needs to solve Equation (15). This can be of higher order, since the expressions $C(G)$ are polynomials in the variables $\hat{\sigma}_i$. However, the

[5] In fact, this holds for any boolean property Q.

perturbations $\sigma_i - \hat{\sigma}_i$ are always bound by $G_i(V)$, the number of subgoals in V that refer to R_i, and often it is OK take $\hat{\Sigma} \approx \Sigma$.

The case of fanout statistics is complicated by the fact that $\hat{\Sigma}$ has a more general form of fanout statistics than we consider in this paper: it needs to allow different σ's for different $\bar{a}$'s in Equation (4) (see Sec. 2). For practical purposes we argue that one should always take $\hat{\Sigma} = \Sigma$.

Perturbations due to FDs We show that the FDs cause even smaller perturbations, i.e. $P_{\Sigma,\Gamma}[Q]$ is very close to $P_\Sigma[Q|\Gamma]$ and becomes equal asymptotically. More formally,

Proposition 3.12. *For any statistics* Γ *and FDs* Σ*,*

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

The intuition behind this is that a data instance chosen randomly from distribution P_Σ satisfies functional dependencies with very high probability: recall from Sec. 2.1 that we have required Γ and Σ to have no conflicts. This is because a functional dependency is the negation of a conjunctive query with $\neq$, to which Theorem 3.2 applies; hence $1 - \mathbf{P}_\Sigma[\Gamma] \leq O(1/n)$. Using this we can prove that adding functional dependencies as additional constraint does not change the statistics asymptotically (although it does change the query probabilities, see Example 3.6).

Based on our discussion, we argue that the entropy maximization distribution $\mu_{\Sigma,\Gamma,V}[-]$ should always be replaced with the conditional binomial distribution $\mu_\Sigma[- \mid V, \Gamma]$. The error is small, and, if we are only interested in probable answers, or almost certain answers, then Proposition 3.8 implies that there is no error at all.

3.3 Query Answering

Given Σ, Γ, V and a query q, we study the following two query answering problems: (1) compute all probable answers, (2) compute all almost certain answers. In the corresponding two decision problems, we have a boolean query Q and ask whether Q is probable, or almost certain respectively. We consider both the query complexity, and the data complexity, when V is obtained from a view instance J: in that case we write $V = V_J$ and study the complexity as a function of J.

It follows from Proposition 3.8 that the problems discussed in this section are independent on the values of the statistics Σ (but do depend on the attribute partitions). Thus, we will omit Σ.

Probabilistic Query Rewriting We will assume $\Gamma = \emptyset$ in our discussion: extensions to $\Gamma \neq \emptyset$ are non-trivial and are deferred to the full paper. We assume arbitrary statistics, but our examples only illustrate cardinality statistics.

Fix a set of views $\bar{v} = (v_1, \ldots, v_m)$, and a query q. Define:

$$\begin{aligned} p(J) &= \{t \mid t \text{ is a probable answer to } q\} \\ ac(J) &= \{t \mid t \text{ is an almost certain answer to } q\} \end{aligned}$$

We are interested in special cases when p and/or ac can be expressed as queries. We call a *probable rewriting* of q a query expression for p, and an *almost certain* rewriting of q a query expression for ac. We note that neither p nor ac need to be monotone as the following example shows:

Example 3.13 Consider the views and the query below:

$$\begin{aligned} v_1(x,y,u) &:- R(x,y,-,u) \\ v_2(y,z) &:- R(-,y,z,-) \\ v_3(y,z,u) &:- R(-,y,z,u) \\ q(x,z) &:- R(x,-,z) \end{aligned}$$

Let $J_1 = \{(a,b,d)\}$, $J_2 = \{(b,c_1)\}$, $J_3 = \emptyset$. Then one can check that (a,c_1) is an almost certain answer to q: $\mu[q(a,c_1)] = 1$. Add the tuple $\{(b,c_2,d)\}$ to J_3, and now $\mu[q(a,c_1)] = 0$. Hence, both $p(J)$ and $ac(J)$ are non-monotone.

Next, we show that, in general, both p and ac need to be at least recursive.

Example 3.14 Consider the view and query below:

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

An instance J for v represents a graph, and we can prove that both $p(J)$ and $ac(J)$ are precisely the transitive closure of J. This is because $MGUQ^0(V_J)$ consists of a single query, representing the connected components of J: by Prop. 3.8, in order for $q()$ to be probable, there must be a homomorphism from $q()$ to V_J, hence a, b are in the same connected component. Hence, both p and ac can be computed by a simple datalog program.

To see why, let $J = \{(m,n),(n,p),(r,s)\}$, then V_J is:

$$V_J :- R(m,z_1), R(n,z_1), R(n,z_2), R(p,z_2), R(r,z_3), R(s,z_3) \qquad (16)$$

and the unique unifier with minimum D is:

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

Clearly both the probable answers, and the almost certain answers to q are $\{(m,n),(n,p),(n,p),(r,s)\}$ and all loops $\{(m,m),(n,n),\ldots\}$.

We show next that, in some restricted cases, p and/or ac can be expressed by a datalog program. In general, this is not possible: this follows from our complexity results below.

Definition 3.15. *A boolean conjunctive query V is* fully unifiable *if there exists some most general unifier h for $V^{(*)}$ s.t. for any two subgoals g, g' in $V^{(*)}$, if they can be unified, then h unifies them (i.e. $h(g) = h(g')$). We call h the full unifier, and $U = h(V)$ the fully unified query.*

Call a conjunctive query V strict *if any two subgoals g, g' in $V^{(*)}$ that unify, have some common constant in the same position, which is a proper position, i.e. contributes to the proper arity of that subgoal.*

Example 3.16 Consider the following three queries:

$$\begin{aligned} V_1 &:- R(a,-,-), R(-,b,-), R(-,-,c) \\ V_2 &:- R(a,b,-)R(-,b,c), R(a,-,c) \\ V_3 &:- R(a,-,-), R(x,x,x), R(-,b,-) \end{aligned}$$

Assume just cardinality statistics, hence $V^{(*)} = V$. V_1 is fully unifiable but not strict; V_2 is fully unifiable and strict; V_3 is not fully unifiable: we can unify any pairs of subgoals, but not all three.

Theorem 3.17. *Let V be fully unifiable and let U be its fully unified query. Then,*

1. *$\mu[Q \mid V] > 0$ iff there $\exists$ homomorphism from Q to U.*
2. *If V is strict, then $\mu[Q \mid V] > 0$ iff $\mu[Q \mid V] = 1$*

Proof.

1. Let U_1 be any query in $MGUQ^0(V)$. Thus, $U_1^{(*)}$ unifies some sub-goals of $V^{(*)}$, and by definition, $U^{(*)}$ can be obtained by further unifying some sub-goals of U_1. By Lemma 3.18, $D(U) \leq D(U_1)$ and hence, U itself must belong to $MGUQ^0(V)$. Now, if there is a homomorphism from Q to U, by Proposition 3.8, $\mu[Q \mid V] > 0$.

   For the other direction, suppose $\mu[Q \mid V] > 0$. Again, by Proposition 3.8, there must exist a query U_2 in $MGUQ^0(V)$ and a homomorphism from Q to U_2. Also, since $U^{(*)}$ can be obtained by unifying some sub-goals of $U_2^{(*)}$, there is a homomorphism from U_2 to U. Thus, there is a homomorphism from Q to U.

2. If V is strict, then $MGUQ^0(V) = \{U\}$. This is because if U_1 is any other query in $MGUQ^0(V)$ then U can be obtained by unifying some sub-goals of U_1. However, since V is strict, $D(U)$ has to be strictly less than $D(U_1)$. This leads to a contradiction as $MGUQ_0(V)$ cannot have queries with different D.

   Hence, for any query Q, $MGUQ^0(QV) \subseteq MGUQ^0(V)$ iff $MGUQ^0(QV) = MGUQ^0(V)$. By Proposition 3.8, $\mu[Q \mid V] > 0$ iff $\mu[Q \mid V] = 1$.

□

Lemma 3.18. *If Q_1 and Q_2 are queries such that $Q_2^{(*)}$ is obtained by unifying two sub-goals of $Q_1^{(*)}$, then $D(Q_2) \leq D(Q_1)$.*

Proof. Suppose the two sub-goals to be unified correspond to the relation $R^{(i)}$ in the extended schema. The unifier equates the variables/constants that occur in the same position of the two sub-goals. Consider the positions that correspond to the attributes $\bar{A}_i$ (recall that these are the attributes that contribute to the proper arity of $R^{(i)}$).

First, assume that the two sub-goals have the same symbols in all the positions except $\bar{A}_i$. Then, after unification, the decrease in the number of distinct variables is at most $|\bar{A}_i|$. Also, after unification, there is exactly one less sub-goal of type $R^{(i)}$, which decreases the total arity by $|\bar{A}_i|$. Thus,

$$D(Q_2) = A(Q_2) - V(Q_2) \leq A(Q_1) - |\bar{A}_i| - (V(Q_1) - |\bar{A}_i|) = D(Q_1)$$

Now consider the other case when the two sub-goals do not have the same symbols in previous positions. Then, the unification can be carried out as a sequence of unifications, all of the first kind as follows: unify the corresponding $R^{(j)}$ ($j \leq i$) sub-goals of the two sub-goals starting from the lowest j where they differ and going up to i. Since $D(Q)$ decreases at each step by the above argument, the lemma holds. □

The following is a necessary and sufficient criterion for V to be fully unifiable. Construct the following graph $G(V)$. The nodes are the variables and constants in V. An edge (u, v) is added whenever there are two subgoals in $V^{(*)}$ that can be unified and the unification equates u to v.

Proposition 3.19. *V is fully unifiable iff there exists no path in $G(V)$ between two different constants.*

The proof is straightforward and omitted. For a simple illustration, the query V_3 in Example 3.16 has two edges (a, x) and (x, b) in the graph, since x is unifiable with both a and b, and this gives us a path from a to b.

Now we can turn our attention to the case when V is derived from a view instance J. We call $\bar{v}$ *fully unifiable* if $\forall J$, V_J is fully unifiable; we call it *strict* if, in addition, $\forall J$, V_J is strict. We describe a necessary and sufficient condition for $\bar{v}$ to be fully unifiable and strict, which can be checked in PTIME on $\bar{v}$.

Let $\bar{v} = \{v_1, \ldots, v_m\}$ and assume that they have disjoint sets of variables (otherise, rename the variables in each view). We construct the following edge-labeled, multi-graph $G(\bar{v})$. The nodes are all the variables and constants occurring in the m views. We describe the edges e and their labels $C(e)$ next. Consider any two pairs of subgoals g, g' that are unifiable, where g is in $v_i^{(*)}$ and g' is in $v_j^{(*)}$. Neither g and g', nor v_i and v_j need to be distinct. Let (x_1, y_1), …, (x_k, y_k) be the pairs of variables and/or constants equated by the unification: for every pair (x_i, y_i) we construct an edge e_i from x_i to y_i. All k labels $C(e_1), \ldots, C(e_k)$ will be the same, and are defined next. Call the pair (x_i, y_i) a *condition* if both x_i and y_i are either a constant or a head variable, and let $D(g, g')$ denote the set of all conditions. Then, all k labels are equal to $D(g, g')$.

The intuition is the following. The nodes in $G(V_J)$ correspond to $|J|$ disjoint copies of the nodes of $G(\bar{v})$. In each copy, the head variables are replaced with constants, hence two subgoals g and g' that are unifiable in $G(\bar{v})$ may no longer be unifiable once the head variables are substituted with constants: the conditions on the edges in $G(\bar{v})$ represent the condition that the edge exists between two such copies.

We use $G(\bar{v})$ as follows. Let $Const$ be the set of constants occurring in all views, and $\Delta = \{(a, a) | a \in Const\}$. First, for each path p in the graph we define a set of conditions $C(p)$. If p is one edge, e, then $C(p)$ is $C(e) \cup \Delta$. If p is obtained by expanding q with one edge e, then $C(p)$ is $C(q) \circ (C(e) \cup \Delta)$ (composition of two binary relations). A path p is *contradictory* if $C(p)$ contains a pair of two different constants, (a, b) with $a \neq b$. Next, we can prove:

Proposition 3.20. *Let $\bar{v}$ be a set of views. The following are equivalent:*

1. *For any view instance J, V_J is fully unifiable.*
2. *For any non-contradictory path p in $G(\bar{v})$ between x and y s.t. both x and y are either head variables or constants, we have $(x, y) \in C(p)$.*

Moreover, the second condition can be checked in PTIME by a dynamic programming algorithm.

Proposition 3.21. *The following are equivalent:*

1. *For any J, V_J is strict.*
2. *Any two unifiable subgoals g, g' of $v_i^{(*)}$ and $v_j^{(*)}$ respectively have a common position where both have a head variable or a constant.*

Thus, the condition still prohibits two distinct constants to be connected by a path (except for contradictory paths, which don't correspond to real paths in $G(V_J)$). In addition, it also prohibits two head variables, or a head variable and a constant to be connected by a path, unless the condition on the path implies that they must be indeed equal.

We explain this construction on the view v in Example 3.14 and the views v_1, v_2 for the running example. Figure 2 (a) shows the graph $G(v)$ for the view v in Example 3.14 and (b) for the views v_1, v_2 in the example in Sec. 1.1. For example, the edge from D_1 to D_2 in (b) represents the fact that $R(N_1, D_1, Z)$ can be unified with $R(X, D_2, B_2)$, but only if the head variable $D_1 = D_2$, since, in V_J these variables are constants. The edge from D_1 to D_1 represents the fact that two different copies of $R(N_1, D_1, Z)$ can also be unified, but only if their head variables N_1 and D_1 are the same; the second edge is for $R^{(1)}(D_1)$.

The graph in Fig. 2 (a) is fully unifiable. For example the edge (X, X) is labeled $X = X$ and is OK. The edge (X, Y) is labeled $X = Y$, again OK. For a longer path, consider $(X, X), (X, Y), (Y, X)$. The edges are labeled $X = X, X = Y, Y = X$ and we can infer that the first X is equal to the last X. The graph in Fig. 2 (b) fails the criterion: the path $(B_2, Z), (Z, B_2)$ is labeled $D_1 = D_2, D_2 = D_1$ and does not imply that $B_2 = B_2$ (these two B_2 refer to the head variables in different copies). Also note that in both the examples, the views are strict.

3.3.1 Recursive Datalog Program

Proposition 3.22. *Given $\bar{v}$ and q there exists a datalog program p, over an instance J, such that:*

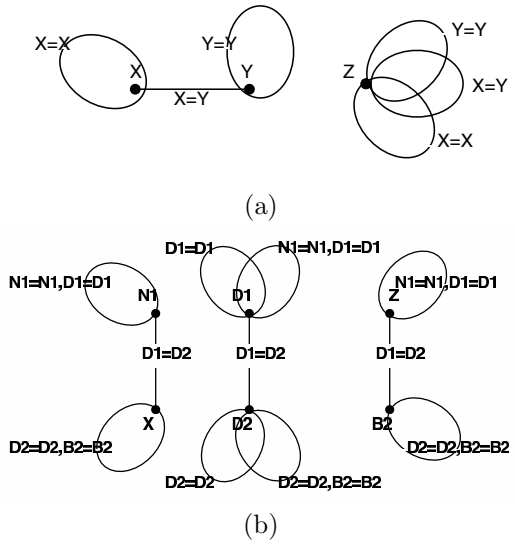

Figure 2: $G(\bar{v})$ for (a) Example 3.14 and (b) Sec 1.1

1. *If $\bar{v}$ are fully unifiable, then p computes the set of all probable answers to q on J.*

2. *If, in addition, $\bar{v}$ are strict, then p computes the set of all almost certain answers to q on J.*

We describe now the boolean datalog program P, given a set of views $\bar{v}$ and a boolean query Q. We call it the *probabilistic rewriting* of Q using $\bar{v}$. The program computes the full unification U of V_J, then checks if there is a mapping from Q to U. The correctness follows from Theorem 3.17. We explain the datalog program in details next.

Consider the view v_i, and let $y_1, y_2, \ldots$ be all variables and constants in v_i. For each tuple t in the view instance v_i^J, the datalog program represents the copy of y_k in $v_i[t/\bar{x}_i]$ as $(i, k, t, pads)$. Here *pads* consists of a repeated dummy constant whose sole purpose is to make all tuples of the same width. The program P operates in three logical steps. First it computes a relation $C(x, c)$ with the association between a variable representations x and the constant c it represents, if any; and it computes the set of all edges in the graph, $E(n, n')$. This part of P is straightforward: in fact, each datalog rule computing C corresponds to a constant or a head variable in some v_i, and each datalog rule computing E corresponds to one edge e in $G(\bar{v})$. Next, P computes the transitive closure of E. This represents the full unifier U: two nodes in V_J are unified in U iff they are in the transitive closure. Finally, P checks if Q can be mapped to U. For that it needs to try all possible ways of unifying subgoals in Q with subgoals in $\bar{v}$ ($|\bar{v}|^{|Q|}$ many ways) and for each it will have one rule, checking the following

conditions: any two occurrences of the same variable in Q are in the transitive closure in E, and any constant in Q is in C.

When q is a non-boolean query and we seek to compute all probable (or almost certain) answers, then the datalog program, denoted p is obtained similarly.

Example 3.23 We illustrate the datalog program p on Example 3.14. There is a single view v and three variables: x, y and z. These are represented as $(1, j, a, b)$ where $j = 1, 2, 3$(for x,y,z) and a,b are values in v. The leading 1 is the index for the view v. Since there is only a single view v, we drop the 1 for simplicity and use (j, a, b) representation. The C table has the following two entries, corresponding to head variables x and y:

$$\begin{aligned} C(1, x, y, x) &:- v(x, y) \\ C(2, x, y, y) &:- v(x, y) \end{aligned}$$

The E table has 6 rules corresponding to the 6 edges in the $G(\bar{v})$ graph shown in Fig 2(a). We illustrate two of these:

$$\begin{aligned} E(1, x, y, 1, x', y') &:- v(x, y), v(x', y'), x = x' \\ E(1, x, y, 2, x', y') &:- v(x, y), v(x', y'), x = y' \end{aligned}$$

Next, the transitive closure:

$$E(i, x, y, i', x', y') \quad :- \quad E(i, x, y, i'', x'', y''), E(i'', x'', y'', i', x', y')$$

Finally, the part matching the query $q() : -R(a, z), R(b, z)$. Both q and v have two sub-goals and they can be paired up in two ways. This corresponds to the following two rules:

$$\begin{aligned} q() &:- C(1, x, y, a), C(2, x', y', b), E(3, x, y, 3, x', y') \\ q() &:- C(2, x, y, a), C(1, x', y', b), E(3, x, y, 3, x', y') \end{aligned}$$

3.3.2 Non-recursive Datalog Program

Call a view V *transitive* if for any two variables x, y if there exists an MGUQ that unifies x and y, then there exists two subgoals g, g' that unify and their MGU equates x and y. The view V_J in Eq (16) is not transitive: U unifies z_1 and z_3, but no two subgoals unify them directly. A set of views $\bar{v}$ is *transitive* if for every J, V_J is transitive. Checking for transitivity is easy. A view V is transitive if in $G(V)$, every two connected nodes have an edge between them. Similarly, a set of views $\bar{v}$ is transitive if in $G(\bar{v})$, for every path p between two nodes, there is also an edge e between the two nodes with $C(e) \subseteq C(p)$.

The views v_1, v_2 in our running example are transitive. If $\bar{v}$ is transitive, then there exists a rewriting P in non-recursive datalog of (a boolean query) Q s.t. $P(J)$ is true iff $\exists U \in MGUQ(V_J)$, $\exists$ homomorphism $h : Q \to U$. To apply

Prop 3.8 we also need U to be of minimal degree. Call a view V *confluent* if any most-general-unifying query $U \in MGUQ(V)$ either has minimum degree, or can further be unified to a minimum-degree unifier $U_0 \in MGUQ^0(V)$. The query V :- $R(-, b, d, e)$, $R(a, b, -, e)$, $R(-, b, c, -)$ is not confluent: the minimum degree is $D = 6$, given by the MGUQ $R(a, b, d, e), R(-, b, c, -)$, but unifying the last two goals gives $R(-, b, d, e), R(a, b, c, e)$ with $D = 7$, and there is no way we can further unify it to get $D = 6$. An example of a confluent query is $V = R(a, b, -)$, $R(-, b, c)$,$R(-, b, d)$. We say that $\bar{v}$ is confluent if for all J, V_J is confluent. In the full paper we give a sufficient condition for $\bar{v}$ to be confluent: in particular, v_1, v_2 in our running example are confluent.

Proposition 3.24. *If $\bar{v}$ is confluent and transitive, then for every q there exists a probable rewriting p of q, in non-recursive datalog.*

The non-recursive datalog is the same program described in Sec 3.3.1 without the transitive closure.

Example 3.25 For our running example in Sec. 1.1 the probable rewriting is:

$$p(N) \quad :- \quad S_1(N, D), S_2(D, \texttt{EE1})$$

3.3.3 Complexity of the Query Answering Problem

We now describe the complexity results.

Theorem 3.26. *(i) The query complexity of deciding whether $\mu[Q|V] > 0$ is* NP*-hard. (ii) The query complexity of deciding whether $\mu[Q|V] = 0$ is* NP*-hard. (iii) There exists a view $v(\bar{x})$ and a query Q such that the data complexity of deciding $\mu[Q|V_J] > 0$ is* NP*-hard.*

Before we give a proof, we need the following result.

Theorem 3.27. *Given a view $v(\bar{x})$ and an integer e, the data complexity of deciding $exp(V[J/x]) \leq e$ is* NP*-complete.*

Proof. Deciding $exp(V[J/x]) \leq e$ is clearly in NP since one just has to guess a substitution Q' with $exp(Q') \leq e$.

To show that the problem is NP-complete, we give a reduction from 3-COLOR, i.e. deciding if a graph can be colored with three colors. The database schema has two tables, E and V. The view V is defined as follows:

$$\begin{aligned} V(k, N_1, N_2) \quad :- \quad & E(k, R, G), E(k, R, B), E(k, G, R), \\ & E(k, G, B), E(k, B, R), E(k, B, G), \\ & E(k, x, y), V(N_1, x), V(N_2, y) \end{aligned} \tag{17}$$

Let G be a graph and we want to test if it can be colored with 3 colors. We construct a certain instance J as follows: for each edge $e = (v_i, v_j)$ we create a tuple $V(e, v_i, v_j)$ in J.

Now we show that $G \in$ 3-COLOR iff $exp(V[J/x]) \leq 18m + 2n$, where m and n are the number of edges and vertices in G respectively.

To calculate $exp(V[J/x])$, we consider the ideal unifier, and contribution from V sub-goals and E sub-goals. For each vertex v_i, at best, all the sub-goals $V(v_i, x)$ can be unified. In this case, a single V sub-goals will remain for each vertex contributing 2 to the total arity. Thus, the total contribution of V sub-goals is $\geq 2m$ and equality holds if x and y values are consistent among each tuple of V. Now we look at E sub-goals. For each tuple $V(e, v_i, v_j)$, the sug-goal $E(e, x, y)$ can at best be unified with one of the other six E sub-goals and sum of arities of E sub-goals for each edge is at least 18. Thus, the total contribution of E sub-goals is $\geq 18m$ and the inequality holds iff for each $V(e, v_i, v_j)$, both x and y are assigned differnent values from the set $\{R, G, B\}$.

Putting together the two results, we have $exp(V[J/x]) \geq 18m + 2n$ and the inequality holds iff each vertex can be assigned a color from $\{R, G, B\}$ such that vertices of same edge have different colors. In other words, $exp(V[J/x]) \leq 18m + 2n$ iff $G \in$ 3-COLOR. This completes the reduction. □

Theorem 3.28. *There exists a view $v(\bar{x})$ and a query Q such that the data complexity of deciding $\mu[Q|V_J] > 0$ is* NP*-hard.*

Proof. Again, we will show a reduction from 3-COLOR.

Let G be a graph and we want to test if it can be colored with 3 colors. Consider a sequence of graphs $G_0, G_1, \cdots G_m$ such that $G_m = G$ and G_i is obtained from G_{i+1} by deleting an edge. Thus, G_0 is the empty graph.

Clearly, G_0 is 3-colorable. Suppose we have established that G_i is 3-colorable for some i. To test G_{i+1}, consider the view definition of Eq 17. Create an instance J for the view using the construction of Theorem 3.27 on graph G_i. Let $e = (v_i, v_j)$ be the edge that is deleted from G_{i+1} to obtain G_i. Define query Q to be $Q() : V(v_i, R), V(v_j, G)$.

We claim that $\mu[Q|V_J] > 0$ iff G_{i+1} is 3-colorable.

Recall we have already established that G_i is 3-colorable. As the proof of Theorem 3.27 shows, each of the unifier that attains the *exp* corresponds to a 3-coloring of G. Also, if G_{i+1} is 3-colorable, there must be a 3-coloring with $v_i = R$ and $v_j = G$ (since we can always rotate colors). If we take this coloring, it is also a 3-coloring of G_i and there is a homomorphism from QV_J to V_J. Thus, $\mu[Q|V_J] > 0$ iff G_{i+1} is 3-colorable.

We can thus test the 3-colorability of G by starting from G_0 and a sequence of above calls. This shows that the data complexity of deciding if $\mu[Q|V_J] > 0$ is NP-hard. □

4 Probabilistic Views

Now we consider the PQAP in its full generality. We are given Γ, Σ, F and a query q, and want to compute the set of pairs $(t, \mu_{\Gamma,\Sigma,F}[t \in q])$. Notice that this is a probabilistic set of answers, i.e. a set of tuples with associated probabilities. When F is derived from views $\bar{v}$ and a probabilistic instance J, we are interested

in fining a rewriting of q that computes this set from J. As before, we will study the boolean case, Q, and derive the non-boolean case. 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 3.10, 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:

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

$$\bar{V}_\Delta = \bigwedge_{j \in \Delta} V_j \wedge \bigwedge_{j \notin \Delta} \neg V_j$$

For each instance I there exists a unique Δ s.t. $V_\Delta(I)$ is true. The following generalizes Lemma 3.10.

Theorem 4.1. *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. $I \models \Gamma$:*

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

where Δ is s.t. $V_\Delta(I)$ is true. For a query Q, it follows:

$$\mathbf{P}_{\Sigma,\Gamma,F}[Q] = \sum_\Delta f C_\Delta \mathbf{P}_{\hat{\Sigma}}[Q\bar{V}_\Delta \mid \Gamma] \tag{18}$$

Proof. (Sketch) To give an essence of the proof, we only illustrate on the case when there is a single relation $R(A_1, \ldots, A_m)$, and Σ contains only a cardinality statistics, $card_R(A_1, \ldots, A_m) = \sigma$.

Let χ_i denote the characteristic function of the view v_i, i.e. $\chi_i(I)$ is 1 iff $I \models V_i$ and 0 otherwise. Represent any distribution $\mathbf{P}$ as a vector $\bar{x}$ of 2^{n^m}

variables $x_I \in [0,1]$, s.t. $\mathbf{P}[I] = x_I$. $\mathbf{P}_\Sigma$ is defined by:

$$\begin{aligned} f_1(\bar{x}) &= \sum_I x_I - 1 = 0 \\ f_2(\bar{x}) &= \sum_I |I| x_I - \sigma = 0 \\ v_i(\bar{x}) &= \sum_I x_I \chi_i(I) = p_i \\ H(\bar{x}) &= \sum (x_I \log 1/x_I) = \text{ is maximized} \end{aligned}$$

By the Lagrange multipliers method, there exists a set of constants λ_1 (for f_1), λ_2 (for f_2), $c_1, \cdots c_m$ (for the views) such that:

$$\forall I. \quad \frac{\partial H}{\partial x_I} + \lambda_1 \frac{\partial f_1}{\partial x_I} + \lambda_2 \frac{\partial f_2}{\partial x_I} + \sum_i c_i \frac{\partial v_i}{\partial x_I} = 0$$

Substituting H, f_1, f_2 and simplifying, we get:

$$\log 1/x_I - 1 + \lambda_1 + \lambda_2 |I| + \sum_i c_i \chi_i(I) = 0$$

Thus, x_I is of the form

$$x_I = A e^{\sum_i c_i \chi_i(I)} B^{|I|}$$

Let $C_i = e^{c_i}$. Also, let Δ be such that $V_\Delta(I)$ is true. Then,

$$x_I = A C_\Delta B^{|I|}$$

Here, the function $AB^{|I|}$ is just a constant times a binomial distribution (although not necessarily the originial binomial distribution). Hence, x_I can be written as $f C_\Delta P_{\Sigma'}(I)$ for some constant f and some statistics Σ'. Thus we have shown that $P_{\Sigma,\Gamma,F}(I) = f C_\Delta P_{\Sigma'}(I)$. This distribution looks like several scaled versions of the same binomial distribution patched together. □

In the sequel we will abbreviate $\mathbf{P}_{\hat{\Sigma}}[- \mid \Gamma]$ simply with $\mathbf{P}[-]$. First we assume that no view is probable given the others (Definition 3.7). That is, for any j, denoting W the conjunction of all views other than V_j, we assume $\mu[V_j \mid W] = 0$. This implies that $\mathbf{P}[Q\bar{V}_\Delta]$ in Eq.(18) 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.(18) becomes Eq.(19) below:

$$\mathbf{P}_{\Sigma,\Gamma,F}[Q] = \frac{\sum_\Delta C_\Delta \mathbf{P}[V_\Delta] \mathbf{P}[Q \mid V_\Delta]}{\sum_\Delta \mathbf{P}[V_\Delta]} \tag{19}$$

Assume for the moment that all probabilistic facts are mutually independent: we prove later that this holds. That is $\mathbf{P}_{\Sigma,\Gamma,F}[V_\Delta] = \prod_{j \in \Delta} \mathbf{P}_{\Sigma,\Gamma,F}[V_j] = p_\Delta$. Substitute $Q = V_{\Delta_0}$ in (19), and note that $\mathbf{P}[V_{\Delta_0} \mid V_\Delta]$ is 1 when $\Delta_0 \subseteq \Delta$ and

≈ 0 otherwise (since $j \notin \Delta$ implies $\mu[V_j \mid V_\Delta] = 0$): this leads to (20) below, which, in turn, leads to (21) using an inclusion-exclusion argument:

$$\forall \Delta_0. \quad p_{\Delta_0} = \frac{\sum_{\Delta_0 \subseteq \Delta} C_\Delta \mathbf{P}[V_\Delta]}{\sum_\Delta \mathbf{P}[V_\Delta]} \tag{20}$$

$$\frac{C_\Delta \mathbf{P}[V_\Delta]}{\sum_\Delta \mathbf{P}[V_\Delta]} = \sum_{\Delta \subseteq \Gamma} (-1)^{|\Gamma - \Delta|} p_\Gamma = \bar{p}_\Delta \tag{21}$$

Substituting back in (19) and taking the limit $n \to \infty$:

Theorem 4.2. *Assuming no view is probable given the others, the solution to the PQAP problem is:*

$$\mu_{\Gamma,\Sigma,F}[Q] \approx \sum_\Delta \bar{p}_\Delta \mu_{\hat{\Sigma}}[Q \mid V_\Delta, \Gamma] \tag{22}$$

Here $\mu_{\hat{\Sigma}}$ corresponds to the binomial distribution for $\hat{\Sigma}$. Moreover the perturbation from $\hat{\Sigma}$ to Σ is bounded by the size of F.

We can now verify the independence assumption, asymptotically: computing $\mu_{\Sigma,\Gamma,F}[V_\Delta]$ with formula (22) gives us indeed p_Δ, since no view is probable given the others. Details are included in the full version.

Example 4.3 Consider our example in Sec. 1.1. Since J has five tuples there are five boolean views, $V_1, V_1', V_2, V_2', V_2''$, with the probabilities listed next to the tuples (0.45, 0.60, etc). Consider the probability that `Larry Big` is an answer to the query: this is the probability of the query $Q :- R(\texttt{LarryBig}, -, \texttt{EE1})$. We will illustrate the query evaluation assuming only the views V_1 and V_2 [6]. Thus, we have four subsets, $\Delta_0 = \phi$, $\Delta_1 = \{V_1\}$, $\Delta_2 = \{V_2\}$ and $\Delta_3 = \{V_1, V_2\}$. The probability of Q is thus:

$$\begin{aligned} \mu_{\Gamma,\Sigma,F}[Q] &= \bar{p}_{\Delta_0} \mu_{\hat{\Sigma}}[Q \mid V_{\Delta_0}, \Gamma]) + \bar{p}_{\Delta_1} \mu_{\hat{\Sigma}}[Q \mid V_{\Delta_1}, \Gamma]) \\ &+ \bar{p}_{\Delta_2} \mu_{\hat{\Sigma}}[Q \mid V_{\Delta_2}, \Gamma]) + \bar{p}_{\Delta_3} \mu_{\hat{\Sigma}}[Q \mid V_{\Delta_3}, \Gamma]) \end{aligned}$$

We have $\mu_{\hat{\Sigma}}[Q \mid V_{\Delta_i}, \Gamma]) = 0$ for $i = 0, 1, 2$ and equal to $1/\sigma_2$ for $i = 3$. Thus,

$$\mu_{\Gamma,\Sigma,F}[Q] = \bar{p}_{\Delta_0} \frac{1}{\sigma_2} = \mathbf{P}[V_1] * \mathbf{P}[V_2] / \sigma_2$$

Hence, the probability of `Larry Big` is $0.45 * 0.25/5$.

Finally, we show that the PQAP problem has a connection to the problem of query evaluation on probabilistic database [8]. A probabilistic database D is a database where each tuple has a probability associated with it. All the tuples are assumed to be independent and this defines a probability distribution over all possible databases. Any query p has a probabilistic semantics on D: it returns

[6] It can be shown that others do not matter here.

a set of pairs $(t, \mathbf{P}[t \in p])$, i.e. a tuple plus the probability that it is an answer to $p(D)$. Some efficient evaluation techniques for SQL queries on probabilistic databases are discussed in [8]. We now show how these techniques can be used to solve the PQAP problem.

In the PQAP problem, we have a query Q and a probabilistic view instance J for a set of views $\bar{v} = v_1, v_2, \ldots$. Recall that J consists of a set of tuples, plus a probability for each tuple. Thus, J can be seen as a probabilitic database. In Sec 3.3, we saw that the query Q often has a rewriting over the views. The next result shows when is it possible to run the rewriting directly on the probabilitic J and retrieve the answers to the PQAP.

Theorem 4.4. *Let $\bar{v}$ be fully unifiable and strict and let as be an almost-certain rewriting of Q using the views $\bar{v}$ (which exists by Proposition 3.22. Then the answer of as on the* probabilistic *data instance J is precisely the set $(t, \mu_{\Gamma,\Sigma,F}[t \in Q])$.*

Proof. Consider any tuple t in the answer to the rewriting as. We will show that it has the same probability under both semantics.

First consider the probabilistic databases semantics[8]. We have a set of possible worlds, each corresponding to a subset of J. Let W_Δ denote the world where J consists of exactly V_Δ. Then, probability of W_Δ is p_Δ. Probability of t is the sum of probabilities of the worlds where it is an answer, which is

$$\sum_{\Delta : t \in as(V_\Delta)} \bar{p}_\Delta \tag{23}$$

Now we consider the PQAP semantics. Let Q_t be the boolean query obtained by substituting the head variables of Q with t. By Theorem 4.2, the probability of Q_t is

$$\mu_{\Gamma,\Sigma,F}[t \in Q] = \sum_{\Delta} \bar{p}_\Delta \mu_{\Sigma'}[Q_t | V_\Delta, \Gamma] \tag{24}$$

Now, since $\bar{v}$ is fully unifiable and strict, $\mu_{\Sigma'}[Q_t|V_\Delta, \Gamma]$ is either 0 or 1. Also, since as is an almost-certain rewriting, $\mu_{\Sigma'}[Q_t|V_\Delta, \Gamma] = 1$ iff $t \in as(V_\Delta)$. Thus, the quantities in Eq 23 and 24 are equal. □

5 Proofs

5.1 Proof of Theorem 3.2

The proof follows from Theorem 5.3 and Corollary 5.11. First, we need some notations.

An *event* is a set of tuples, $e \subseteq Tup$, and we denote $\mu_n[e]$ the probability that all tuples are in a randomly chosen database instance. If $e_1, \ldots, e_m$ are

events then $e_1 \vee \ldots \vee e_m$ denotes the event that at least one of them happens, i.e. a randomly chosen database instance contains all tuples in e_i, for some $i = 1, \ldots, m$. The proof of theorem 3.2 relies on the following inequalities, representing a lower bound and an upper bound for $\mu_n[e_1 \vee \ldots \vee e_m]$, and which are standard in probability theory:

$$\sum_{i=1,m} \mu_n[e_i] - \sum_{1 \leq i < j \leq m} \mu_n[e_i e_j] \quad \leq \quad \mu_n[e_1 \vee \ldots \vee e_m] \tag{25}$$

$$\sum_{i=1,m} \mu_n[e_i] \quad \geq \quad \mu_n[e_1 \vee \ldots \vee e_m] \tag{26}$$

The event $e_i e_j$ represents the fact that all tuples in e_i and e_j are chosen; it is equivalent to the event $e_i \cup e_j$.

Given a conjunctive query Q_0, denote $Q_0^{\neq}$ the query obtained by adding all possible $\neq$ predicates, between any two distinct variables in Q_0, and between any variable and constant in Q. For example, if $Q_0 \leftarrow R(a, x), R(x, y)$ then $Q_0^{\neq} \leftarrow R(a, x), R(x, y), x \neq y, x \neq a, y \neq a$. Let $MUQ(Q)$ denote the set of all minimal unifying queries of Q, i.e. queries which are minimal and of the form $h(Q)$ where h is a substitution on query Q. Note that we do not include two queries in $MUQ(Q)$ that are identical up to variable renaming. Thus, any two distinct queries in $MUQ(Q)$ are non-isomorphic.

The proof of the main result relies on the two inequalities in the following Lemma, and applying Eq.(25) and (26) to each of them.

Lemma 5.1. *For any conjunctive query Q,*

$$Q \quad \equiv \quad \bigvee \{Q_0^{\neq} \mid Q_0 \in MUQ(Q)\} \tag{27}$$

Proof. The containment in one direction is easy: $Q_0 \subseteq Q$ for $Q_0 \in MUQ(Q)$ follows from the standard homomorphism theorem (since $Q_0 = \eta(Q)$), and $Q_0^{\neq} \subseteq Q_0$ is also immediate. For the other direction, consider one database instance I where Q is true, and let θ be the substitution that makes Q true. We will find some $Q_0 \in MUQ(Q)$, s.t. $Q_0^{\neq}$ is also true in I. Let $const(Q)$ be all constants in Q, and $C = \{c_1, \ldots, c_m\}$ be all constants in $\theta(Q)$ that are not in $const(Q)$. Let $z_1, \ldots, z_m$ be m fresh variables, one for each constant in C. Define the following substitution η on Q's variables. If $\theta(x) \in const(Q)$, then $\eta(x) = \theta(x)$; otherwise, if $\theta(x) = c_i$, $i = 1, \ldots, m$, then $\eta(x) = z_i$. Let $Q_0' = \eta(Q)$. Let Q_0 be a query formed by a subset of sub-goals of Q_0' such that Q_0 is minimal. Q_0 can be expressed as $\tau(Q_0')$ where τ is the substitution that maps the redundant sub-goals of Q_0' to the minimal part. Thus Q_0 must belong to $MUQ(Q)$ since $Q_0 = \tau(\eta(Q))$ and it is minimal. The valuation θ_0 defined by $\theta_0(z_i) = c_i$, $i = 1, m$ is defined on $Q_0^{\neq}$, and $\theta_0(Q_0) = \theta(Q)$, proving that $Q_0^{\neq}$ is true on the instance I. □

Upper bound Here we establish the upper bound of Theorem 3.2. First we need the following lemma. Recall the definition of a free variable from Sec 3.1.2.

Lemma 5.2. *Let Q be any conjunctive query where every variable is a free variable. Then,*

$$\mu_n(Q^{\neq}) = C(Q)/n^{D(Q)}$$

Proof. Denote $Q^{(i)}$ the query consisting of all the subgoals in $Q^{(*)}$ that refer to relation $R^{(i)}$. Let $T^{(i)}$ denote the set of trivial sub-goals in $Q^{(i)}$. We have

$$\mu_n(Q) \quad = \quad Q^{(1)} \prod_{2 \leq i \leq n} \mu_n[Q^{(i)}|Q^{(i-1)}]$$

Now, given that $Q^{(i)}$ is true, each of the non-trivial sub-goals g in $Q^{(i+1)}$ as well as $T^{(*)}$ are independent. Now we just need to plug in the definitions of $C(Q)$ and $D(Q)$. For a non-trivial goal g, $\mu_n(g) = C(g)n^{D(g)}$. Similarly, the probability that $T^{(*)}$ is true is given by Eq (9). Putting together everything, we get the desired result. □

Now we prove the upper bound.

Theorem 5.3. *For any conjunctive query Q,*

$$\mu_n[Q] \quad \leq \quad \frac{\text{coeff}(Q)}{n^{\exp(Q)}} + O(\frac{1}{n^{\exp(Q)+1}})$$

Proof. Let Θ denote the set of all partial substitutions θ defined on a query $Q_0^{\neq}$ that only maps the non-free variables of Q_0 to constants from the domain. Let f be the number of free variables in $Q_0^{\neq}$. For any $\theta \in \Theta$, $\theta(Q_0^{\neq})$ results in a query where all variables are free variables. We have $A(\theta(Q_0^{\neq})) = A(Q_0)$ and $V(\theta(Q_0^{\neq})) = f$. Thus, $D(\theta(Q_0^{\neq})) = A(Q_0) - f$. Also, by the standard semantics of the conjunctive queries, we have

$$Q_0^{\neq} \quad \equiv \quad \bigvee_{\theta \in \Theta} \{\theta(Q_0^{\neq})\} \tag{28}$$

We apply the upper bound in (26) twice: first to Eq.(27), then, for each unifying query $Q_0 \in MUQ(Q)$, to Eq.(28). We obtain:

$$\mu_n[Q] \quad \leq \quad \sum_{Q_0 \in MUQ(Q)} \sum_{\theta \in \Theta} \mu_n[\theta(Q_0^{\neq})]$$

For each θ that is defined on $Q_0^{\neq}$, we have $\mu_n[\theta(Q_0^{\neq})] = C(Q_0)/n^{A(Q_0)-f}$ by Lemma 5.2 and the facts that $D(\theta(Q_0^{\neq})) = A(Q_0) - f$ and $C(\theta(Q_0^{\neq})) = C(Q_0)$. Moreover, since there are $V(Q_0) - f$ variables that are not free, there are $n^{V(Q_0)-f} - O(n^{V(Q_0)-f-1})$ substitutions θ that are defined on $Q_0^{\neq}$. Hence, for each unifier Q_0, the inner sum above is $C(Q_0)/n^{D(Q_0)} - O(1/n^{D(Q_0)+1})$. When summing up over all unifiers, the dominant terms are those with the lowest $D(Q_0)$, hence we have:

$$\mu_n[Q] \quad \leq \quad \frac{\mathit{coeff}(Q)}{n^{\mathit{exp}(Q)}} + O(\frac{1}{n^{\mathit{exp}(Q)+1}})$$

This establishes the upper bound. □

Lower bound This is harder, because we have to prove that the second order terms in the lower bound of Eq.(25) are negligible: more precisely we show that the total contribution of these terms is $O(1/n^{exp(Q)+1})$. We first apply the lower bound to Eq.(27). The second order terms are here expressions of the form $\mu_n[Q_0^{\neq}Q_1^{\neq}]$, where $Q_0, Q_1 \in MUQ(Q)$. Here $Q_0^{\neq}Q_1^{\neq}$ represents the conjunction of the two boolean queries, and is obtained by first renaming all variables in Q_0 and Q_1 to make them disjoint, and then taking the union of all predicates in the two queries, both subgoals and $\neq$ predicates. The number of such expressions depends only on Q, not on n, so it suffices to show that each such expression is $O(1/n^{exp(Q)+1})$. This follows from Corollary 5.8. We need the following lemmas.

Lemma 5.4. *Let Q_0 be any query in $MUQ(Q)$. Then, $D(Q_0) \geq \exp(Q)$.*

Proof. For Q_0 in $MGUQ(Q)$, the lemma follows from the definition of $exp(Q)$. For any general Q_0 in $MUQ(Q)$, let η be the substitution such that $Q_0 = \eta(Q)$. Consider the partition of sub-goals induced by η on $Q^{(*)}$ and let η' be the most general unifier for the same partition. Thus, there exists a substitution f such that $\eta(Q^{(*)}) = f(\eta'(Q^{(*)})$. Since $\eta(Q^{(*)})$ and $\eta'(Q^{(*)})$ have the same number of sub-goals, $A(\eta(Q)) = A(\eta'(Q))$. Also, the substitution f cannot increase the number of distinct variables and hence $V(\eta(Q)) \leq V(\eta'(Q))$. Thus, $D(\eta(Q)) \geq D(\eta'(Q))$. Now, since η' is a most general unifier, $\eta'(Q) \in MGUQ(Q)$ and hence, $D(\eta'(Q)) \geq exp(Q)$. Putting together everything, $D(Q_0) = D(\eta(Q)) \geq D(\eta'(Q)) \geq exp(Q)$. □

Recall from Sec 3.1.2 that a sub-goal of $Q^{(*)}$ is called trivial if all of its proper attributes have free variables.

Lemma 5.5. *Let Q be any conjunctive query without trivial sub-goals. Let Q' be a query formed by taking a subset of the sub-goals of $Q^{(*)}$. Then*

$$D(Q) \geq D(Q')$$

The inequality is strict iff Q' does not contain at least one non-trivial sub-goal of $Q^{(*)}$.

Proof. We keep adding sub-goals to Q' till we reach Q and count the increase in the arity as well as the number of distinct variables.

Consider the smallest i such that a sub-goal corresponding to relation $R^{(i)}$ is not in Q' and add one such sub-goal to Q'. The true arity of this sub-goal is $|\bar{A}_i|$. Also, the number of new variables this sub-goal can introduce is at most $|\bar{A}_i|$ because a symbol in every other position also appears in a $R^{(i-1)}$ sub-goal in Q'. Hence, the increase in the number of distinct variables is at most the increase in arity. Thus, after adding all the remaining sub-goals, we get $D(Q) \geq D(Q')$. For the equality to hold, all the added sub-goals must have free variables as proper attributes, i.e. they must all be trivial. □

We call a subgoal $R(\cdots)$ of Q *completely trivial* if each of the $R^{(i)}(\cdots)$ subgoals in $Q^{(*)}$ corresponding to this sub-goal are trivial. Thus, a completely

trivial sub-goal has no constants and all the variables are distinct and do not occur elsewhere in the query.

Lemma 5.6. *Let Q be any minimal conjunctive query without completely trivial sub-goals. Let Q_1 be a query formed by taking a subset of the sub-goals of Q. Then*

$$D(Q) \geq D(Q_1)$$

Also, the equality holds iff $Q = Q_1$.

Proof. The inequality follows from Lemma 5.5 by putting $Q' = Q_1^{(*)}$.

Clearly, when $Q = Q_1$, the equality holds. We only have to show that the inequality is strict when Q and Q' differ. It is sufficient to prove this for the case when Q and Q' differ by a single sub-goal.

Suppose Q' is obtained from Q by deleting the sub-goal $g = R(\bar{a}_1, \bar{a}_2 \cdots \bar{a}_k)$ and $D(Q) = D(Q_1)$. The sub-goal contributes to the following k sub-goals in $Q^{(*)}$: $g_i = R^{(i)}(\bar{a}_1, \bar{a}_2, \cdots, \bar{a}_i)$ for $1 \leq i \leq k$. $Q_1^{(*)}$ already contains some of these sub-goals and by Lemma 5.5, all other sub-goals are trivial. Thus, $Q_1^{(*)}$ must contain at least one of these sub-goals, otherwise g would be a completely trivial sub-goal, contradicting our assumption. Let j be the largest i such that $Q_1^{(*)}$ contains g_i. Then, in fact, $Q_1^{(*)}$ contains sub-goals g_1 to g_j. Also, sub-goals g_{j+1} to g_k are all trivial. Thus, positions $\bar{a}_{j+1}, \bar{a}_{j+2} \cdots, \bar{a}_k$ must have fresh and distinct variables. Also, since $Q_1^{(*)}$ contains g_j, Q_1 must contain a sub-goal of the form $R(\bar{a}_1, \bar{a}_2 \cdots, \bar{a}_j, -, - \cdots, -)$. This makes g a redundant sub-goal in Q, contradicting the minimality of Q. Hence, Q and Q_1 are equivalent. □

Lemma 5.7. *Let Q_0 and Q_1 be two minimal conjunctive queries without $\neq$ predicates such that (1) they do not contain any completely trivial sub-goals and (2) they are not isomorphic. Then:*

$$\exp(Q_0^{\neq} Q_1^{\neq}) \quad \geq \quad \min(D(Q_0), D(Q_1)) + 1$$

Proof. Assume the contrary, that $exp(Q_0^{\neq} Q_1^{\neq}) \leq D(Q_0)$ and $exp(Q_0^{\neq} Q_1^{\neq}) \leq D(Q_1)$.

We have the following inequalities:

$$exp(\eta(Q_0^{\neq} Q_1^{\neq})) \geq D(\eta(Q_0^{\neq} Q_1^{\neq})) \geq D(\eta(Q_0^{\neq})) \geq D(Q_0) \tag{29}$$

The first inequality holds by definition of *exp*. The second inequality follows from Lemma 5.6 and the fact that $goals(\eta(Q_0^{\neq}) \subseteq goals(\eta(Q_0^{\neq} Q_1^{\neq})$. The third inequality follows from the following argument: $(\eta(Q_0^{\neq}))^{(*)}$ and $Q_0^{(*)}$ have the same number of sub-goals (since η cannot unify two sub-goal of $Q_0^{\neq}$ because of the inequalities). Hence, $A(\eta(Q_0^{\neq})) = A(Q_0)$. Also, since η cannot increase the number of distinct variables, $V(\eta(Q_0^{\neq})) \leq V(Q_0)$.

Given our first assumption ($exp(\eta(Q_0^{\neq} Q_1^{\neq})) \leq D(Q_0)$), Eq. 29 must have equalities in all the three places. For $D(\eta(Q_0^{\neq}))$ and $D(Q_0)$ to be equal, η must

map variables in $Q_0^{\neq}$ to distinct variables. Thus, $\eta(Q_0^{\neq})$ is isomorphic to $Q_0^{\neq}$. Similarly, by Lemma 5.6, $D(\eta(Q_0^{\neq}Q_1^{\neq})) = D(\eta(Q_0^{\neq}))$ implies that $\eta(Q_0^{\neq}Q_1^{\neq})$ is equal to $\eta(Q_0^{\neq})$, and hence, isomorphic to $Q_0^{\neq}$. A similar argument shows that $\eta(Q_0^{\neq}Q_1^{\neq})$ is isomorphic to $Q_1^{\neq}$. Thus Q_0 and Q_1 are isomorphic, contradicting our assumption. □

We get the following corollary.

Corollary 5.8. *For $Q_0, Q_1 \in MUQ(U)$*

$$\mu_n(Q_0^{\neq}, Q_1^{\neq}) = O(1/n^{\exp(Q)+1})$$

Proof. Both Q_0 and Q_1 are minimal by definition of $MUQ(Q)$. Suppose Q_0 contains a completely trivial sub-goal g referring to a relation R. Thus, g contains all distinct variables disjoint from Q_0. Since Q_0 is minimal, the relation R does not occur in any other sub-goal of Q_0, otherwise g would be a redundant sub-goal. We next show that every query in $MUQ(Q)$ contains g. Consider all the sub-goals in Q that refer to relation R. Since all of them get unified to g in Q_0, which is completely trivial, each one of them must be an isomorphic copy of g and any subset of them also unify to g. Thus, all query in $MUQ(Q)$ contain g, and by their minimality, contain g as a completely trivial sub-goal.

It follows that the set of completely trivial sub-goals is the same for all queries in $MUQ(Q)$. Let q_0 and q_1 denote the queries obtained from Q_0 and Q_1 by stripping off the completely trivial sub-goals. Since Q_0 and Q_1 are non-isomorphic, q_0 and q_1 are also non-isomorphic. Also, $D(Q_0) = D(q_0)$ and $D(Q_1) = d(q_1)$.

From lemmas 5.4 and 5.7, Thus, we have

$$\begin{aligned} exp(q_0^{\neq}, q_1^{\neq}) &\geq min(D(q_0), D(q_0)) + 1 && (30)\\ &= min(D(Q_0), D(Q_1)) + 1 && (31)\\ &\geq exp(Q) + 1 && (32) \end{aligned}$$

Eq (30) follows from Lemma 5.7 and Eq (32) follows from Lemma 5.4.

Thus, $\mu_n(Q_0^{\neq}, Q_1^{\neq}) \leq \mu_n(q_0^{\neq}, q_1^{\neq}) \leq O(1/n^{exp(q_0^{\neq}, q_1^{\neq})+1}) = O(1/n^{exp(Q)+1})$.

The second inequality above follows from our already proven upper bound in Theorem 5.3. □

Corollary 5.9. *For a conjunctive query Q,*

$$\mu_n[Q] \geq \sum_{Q_0 \in MUQ(Q)} \mu_n[Q_0^{\neq}] - O(1/n^{\exp Q+1}) \quad (33)$$

Given the upper bound, it suffices to consider in the sum only those unifiers Q_0 in $MUQ(Q)$ for which $D(Q_0) = exp(Q)$: the others result in lower order terms. We apply now Eq.(28) to $Q_0^{\neq}$, and then the lower bound in (25). The higher order terms are now of the form $\mu_n[\theta(Q_0^{\neq})\theta'(Q_0^{\neq})]$, and we will show that their combined effect is $O(1/n^{exp(Q)+1})$. The number of such terms is now dependent on n so it is not enough to show that each one of them is separately negligible.

Theorem 5.10. *For a conjunctive query Q and $Q_0 \in MUQ(Q)$,*

$$\sum_{\theta,\theta' \in \Theta} \mu_n[\theta(Q_0^{\neq})\theta'(Q_0^{\neq})] = O(1/n^{\exp(Q)+1})$$

Proof. Recall that θ and θ' are partial substitutions that only map the non-free variables of $Q_0^{\neq}$ to constants from the domain. Let f denote the number of free variables in $Q_0^{\neq}$.

Denote $e = \theta(Q_0^{\neq})$ and $e' = \theta'(Q_0^{\neq})$. Both e and e' have both the same number of sub-goals, namely equal to the number of subgoals in Q_0, because both θ and θ' are injective (due to the $\neq$ predicates). We examine their overlap. Consider all subgoals in $Q_0^{(*)}$ that are mapped to the same sub-goal by θ and θ'. Define a new boolean query Q_1 consisting of precisely these subgoals; hence $goals(Q_1) \subset goals(Q_0^{(*)})$ (we cannot have equality because $\theta \neq \theta'$). In fact, that must be a non-trivial sub-goal in $goals(Q_0^{(*)}) \setminus goals(Q_1)$ (since θ and θ' must map at least one non-free variable differently). Hence, by Lemma 5.5, $D(Q_1)$ must be strictly less than $D(Q_0)$.

Now, the intuition is that, when Q_1 has few subgoals (or, e.g., is empty), then $\mu_n[ee']$ is very small, since e and e' are largely independent; when Q_1 has many subgoals, then we use the fact that there cannot be too many pairs of valuations θ, θ' that agree on all subgoals in Q_1. For these we need the following inequalities, which are easily checked. (1) $\mu_n[\theta(Q_0^{\neq})\theta'(Q_0^{\neq})] = O(1/n^{2A(Q_0)-A^*(Q_1)-2f})$, and (2) the number of pairs of substitutions θ, θ' which agree precisely on the subgoals in Q_1 is $O(n^{2(V(Q_0)-V^*(Q_1))-2f})$. Now we can add the second order terms and obtain:

$$\begin{aligned}
\sum_{\theta,\theta' \in \Theta} \mu_n[\theta(Q_0^{\neq})\theta'(Q_0^{\neq})] &= \sum_{Q_1 : goals(Q_1) \subset goals(Q_0)} O(\frac{n^{2(V(Q_0)-V^*(Q_1))-2f}}{1/n^{2A(Q_0)-A^*(Q_1)-2f}}) \\
&= \sum_{Q_1 : goals(Q_1) \subset goals(Q_0)} O(\frac{1}{n^{2D(Q_0)-D^*(Q_1)}}) \\
&= O(1/n^{D(Q_0)+1}) \quad (34) \\
&= O(1/n^{exp(Q)+1}) \quad (35)
\end{aligned}$$

Eq (34) follows from Lemma 5.5 and Eq (35) follows from Lemma 5.4. □

Corollary 5.11. *For any conjunctive query Q,*

$$\mu_n[Q] \geq \frac{\text{coeff}(Q)}{n^{\exp(Q)}} - O(\frac{1}{n^{\exp(Q)+1}})$$

The above corollary, along with Theorem 5.3 proves Theorem 3.2.

Next, we prove Prop 3.11. We first state a generalization of Theorem 3.2. Let X be any function from the set of all possible data instances to real numbers. The expected value of X under the distribution μ_n is given by $E_n(X) =$

$\sum_I X(I)\mu_n(I)$. Similarly, the expected conditional value of X given a boolean query Q is given by $E_n(X|Q) = \sum_I X(I)\mu_n(I|Q)$. Define $E(X)$ as $lim_{n\to\infty}E_n(X)$ and $E(X|Q) = lim_{n\to\infty}E_n(X|Q)$, if the limit exists.

Theorem 5.12. *If $E_n(X|Q)$ converges for all conjunctive queries Q, then*

$$E(X|Q) = \frac{1}{\mu_n(Q)} \sum_{Q_0 \in MGUQ} E(X|Q_0^{\neq})\mu_n(Q_0^{\neq})$$

Proof. We use the following standard results from the probability theory. If $q = q_1 \vee \cdots \vee q_m$,

$$E_n(X|q) \leq \frac{1}{\mu_n(q)} \sum_i E(X_n|q_i)\mu_n(q_i) \quad (36)$$

$$E_n(X|q) \geq \frac{1}{\mu_n(q)} (\sum_i E(X_n|q_i) - \sum_{1\leq i\leq j\leq m} E(X_n|q_i,q_j)\mu_n(q_i,q_k)) \quad (37)$$

We apply the above inequalities to Eq (27). Since $E_n(X|q)$ converges for all q, let $C = max_{Q_0\in MUQ(Q)}E(X|q)$. Hence, for all sufficiently large n and $Q_0 \in MUQ(Q)$, we have $E_n(X|Q_0) \leq 2C$, i.e. less than a constant. Also, for $Q_0, Q_1 \in MUQ(Q)$, we have shown that $\mu_n(Q_0^{\neq}, Q_1^{\neq}) = O(1/n^{exp(Q)})$. Thus, the second order terms in Eq (37) are negligible. Hence, we have

$$E_n(X|Q) = \frac{1}{\mu_n(Q)} \sum_{Q_0 \in MUQ} E_n(X|Q_0^{\neq})\mu_n(Q_0^{\neq}) + O(1/n)$$

Further, only those Q_0 that belong to $MGUQ^0(Q)$ have non-negligible contribution in the above sum. Taking the limit, we get the required result. □

From now on, we assume a single relation R and a single cardinality constraint that the expected size of R is σ. Thus μ_n is the binomial distribution with parameter σ. Also, the following holds for this case: every minimal query is either the trivial query $R(x_1, x_2, \cdots)$ (that asks if R is non-empty) or does not contain any trivial sub-goals.

We analyze a particular function, $Size$. Given a data instance I, $Size(I)$ is simply the number of tuples in I. Recall that given a query Q, $G(Q)$ denotes the number of sub-goals in Q.

Theorem 5.13. *For any minimal query Q that does not contain trivial sub-goals,*

$$E[Size|Q^{\neq}] = \sigma + G(Q)$$

Proof. We apply the inequalities in Eq (36) and (37) to Eq (28). Note that since Q does not contain any free variables, θ consists of full substitutions. Thus, for

each θ, $\theta(Q^{\neq})$ represents a set of $G(Q)$ distinct tuples. One can easily show that $E[Size|\theta(Q^{\neq})] = \sigma + G(Q)$. From Eq (36),

$$\begin{aligned} E_n[Size|Q] &\leq \frac{1}{\mu_n(Q)} \sum_{\theta} E[Size|\theta(Q^{\neq})]\mu_n(Q^{\neq}) \\ &= \frac{\sigma + G(Q)}{\mu_n(Q)} \sum_{\theta} \mu_n(Q^{\neq}) \\ &\leq \sigma + G(Q) \end{aligned}$$

Similarly, from Eq (36),

$$\begin{aligned} E_n[Size|Q] &\geq \sigma + G(Q) - \frac{(\sigma + G(Q))^2}{\mu_n(Q)} \sum_{\theta,\theta'} \mu_n(\theta(Q^{\neq}), \theta'(Q^{\neq})) \\ &= \sigma + G(Q) - O(1/n) \end{aligned}$$

The last equality follows from Theorem 5.10. By taking the limit, we get $E[Size|Q^{\neq}] = \sigma + C(G)$. □

Theorem 5.14. *If Q is any query other than the trivial query,*

$$E[Size|Q] = \sigma + \frac{\sum_{Q_0 \in MGUQ^0(Q)} C(Q)G(Q)}{\sum_{Q_0 \in MGUQ^0(Q)} C(Q)}$$

Proof. If Q_0 is any query in $MGUQ^0(Q)$, it is minimal. Since Q is not the trivial query, Q_0 cannot contain trivial sub-goals. The theorem now follows directly from Theorems 5.12 and 5.13. □

6 Related Work

Several models of probabilistic databases [6, 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.

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 developed a framework for representing complex probabilistic databases, to be used in data integration scenarios. In a LAV approach, the probability distribution on the global instance is given indirectly, through probabilities on the view instances. This allows us to express rich correlations between tuples. Our model also takes as input statistics over the data. Despite its richness, we have shown that the query answering problem is decidable, and sometimes tractable. At an extreme, the query's probabilistic answers can be computed by evaluating a single rewritten query on probabilistic instance: efficient techniques for this are discussed in [8]. Future research is needed in two directions. One is to address some restrictions we impose on statistics: we currently require the statistics on a relation to form a chain, while more general statistics are likely to occur in practice. The second is to study other cases under which PQAP is tractable.

References

[1] Serge Abiteboul and Oliver M. Duschka. Complexity of answering queries using materialized views. In *PODS*, pages 254–263, 1998.

[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] Jihad Boulas, Nilesh Dalvi, Bhushan Mandhani, Shobhit Mathur, Chris Re, and Dan Suciu. Mystiq: A system for returning probabilistic answers to hard queries. In *University of Washington Technical Report*, 2004.

[6] 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.

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

[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.

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