General Database Statistics Using Entropy Maximization

Raghav Kaushik[1], Christopher Ré[2], and Dan Suciu[2]

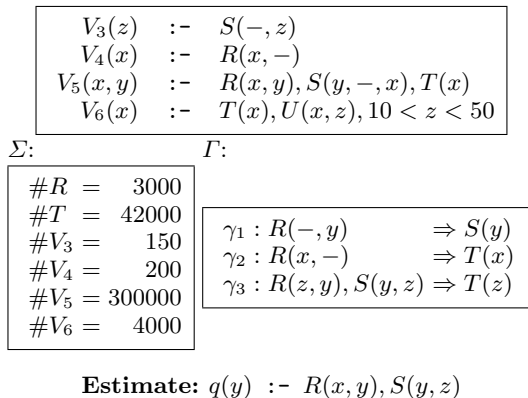[1] Microsoft Research
[2] University of Washington, Seattle WA

Abstract. We propose a framework in which query sizes can be estimated from arbitrary statistical assertions on the data. In its most general form, a statistical assertion states that the size of the output of a conjunctive query over the data is a given number. A very simple example is a histogram, which makes assertions about the sizes of the output of several range queries. Our model also allows much more complex assertions that include joins and projections. To model such complex statistical assertions we propose to use the Entropy-Maximization (EM) probability distribution. In this model any set of statistics that is consistent has a precise semantics, and every query has an precise size estimate. We show that several classes of statistics can be solved in closed form.

1 Introduction

Modern database query optimizers are the result of thousands of man years worth of work from very talented individuals, and so are extremely sophisticated. Although the optimizers themselves are sophisticated, the heuristics they employ are often not: When estimating the selectivity of two predicates, the optimizer might make an *independence assumption* and so, assume that the selectivities of two predicates are independent. When estimating the size of the intersection of two columns, the optimizer might make a *containment assumption* and assume that the values in one column are contained in the other. The cleverness of the optimizer is in *how* and *when* to apply these rules. In spite of this intense effort and the fundamental importance of the problem, there is no general theory that explains how the optimizer should make these choices or when such choices are consistent. In this paper, we take a first step towards such a general, principled theory of how optimizers should make use of statistics.

The lack of a principled framework is likely to become an even more critical problem as new sources of statistics become available to the query optimizer. For example, several proposals [3,17] advocate acquiring *query feedback* and incorporating this statistical feedback into the optimizer. It is easy to collect cardinality statistics from each query as it is executed by the database engine. The difficulty lies in combining this newfound plethora of statistics to produce a single, principled estimate. The lack of such a principled framework is a major reason that execution feedback has not been widely adopted in commercial database engines.

The key object of our study is a *statistical program*, which is a set of pairs (v, d), where v is a query (also called a view) and $d > 0$ is a number. Each pair

$$
\begin{array}{lll}
V_3(z) & \text{:-} & S(-, z) \\
V_4(x) & \text{:-} & R(x, -) \\
V_5(x, y) & \text{:-} & R(x, y), S(y, -, x), T(x) \\
V_6(x) & \text{:-} & T(x), U(x, z), 10 < z < 50
\end{array}
$$

Σ: Γ:

$$
\begin{array}{rr}
\#R & = 3000 \\
\#T & = 42000 \\
\#V_3 & = 150 \\
\#V_4 & = 200 \\
\#V_5 & = 300000 \\
\#V_6 & = 4000
\end{array}
$$

$$
\begin{array}{ll}
\gamma_1 : R(-, y) & \Rightarrow S(y) \\
\gamma_2 : R(x, -) & \Rightarrow T(x) \\
\gamma_3 : R(z, y), S(y, z) & \Rightarrow T(z)
\end{array}
$$

Estimate: $q(y)$:- $R(x, y), S(y, z)$

Fig. 1. An example of a Statistical Program and a query, q whose cardinality we would like to estimate.

(v, d) is called a *statistical assertion*, and means intuitively that the answer to v has expected size d; we write it as $\#v = d$ and say simply that *"the size of v is d"*. A statistical program encodes the information that is available to an optimizer. The primary use of this information is to estimate the expected result size of other queries during query optimization.

Example 1. Figure 1 illustrates a statistical program that asserts the sizes of 4 views and 2 base relations. The program also asks for the estimated size of another query. This program asserts that the expected number of distinct tuples in the relation R is 3000. The program also asserts (via V_3) that the second attribute of S contains 150 values. Our proposal also allows complex assertions that involve joins and arithmetic predicates, such as V_6. In addition to statistical assertions, our model also allows specifies *inclusion constraints*. These constraints are *hard constraints* and must hold in *every instance I* the distribution considers possible, i.e., for which $\mathbf{P}[I] > 0$. For example, γ_1 says that each value in the second column of R is also in the S relation.

In this paper, we study the probability distribution over database instances that satisfy a given statistical program Σ.

1.1 Our Approach: Entropy Maximization

In this paper we define a model for a statistical program using the Entropy Maximization principle. We assume that the relations in the database are drawn from a finite domain D, of size N, and that each database instance I has some probability $\mathbf{P}(I)$. $\mathbf{P}$ is chosen to fit the statistics Σ and without making any other assumptions on the data. More precisely (1) for each assertion $\#v = d$ in Σ, the expected size of v under $\mathbf{P}$ is d, and (2) the probability distribution $\mathbf{P}$ has the maximum entropy among those that satisfy (1) (formal definition given in Sec. 3). The EM model for a statistical program Σ is an instance of the

general EM principle in probability theory, discussed for example by Jaynes [12, Ch.9,11], and which has also been applied to consistent use and construction of histograms [14, 16].

The EM framework has many attractive features. First, any combination of statistical assertions has a well-defined semantics (except, of course, when it is inconsistent); thus, a statistical program is treated as a whole, as opposed to a set of separate synopses. Second, every query has a well defined cardinality estimate; there is no restriction on the query, and the query estimate no longer depends on which heuristics are used to do the estimation. A third reason is that the EM framework has an interesting property that allows us to add a new statistical assertion smoothly: if the estimate of a query q under a statistical program Σ is d, then after adding the assertion $\#q = d$ the new EM probability distribution is identical to the previous one. In practice this means that if we add a small correction to the model, $\#q = d'$ where $d' \approx d$, then the model will change smoothly. The final, and most important conceptual reason is that in a precise sense the probability distribution given by entropy maximization depends *only on the provided statistics* and makes no additional assumptions beyond this. We return to this point when we formally define our model and state its properties in Sec. 3.1.

In this paper, we study the following *model computation problem*: given a statistical program Σ and a set of full inclusion constraints Γ, find a solutions to the EM model. (We explain below the reason for introducing constraints.) Since an EM solution is tied to the particular domain D, we seek to remove the dependency by letting the domain size N grow to infinity as is done, for example in random graphs [9], knowledge representation [2], or asymptotic query probability [5]. Since we seek an analytic understanding of the model, our goal is to find analytic (asymptotic) solutions to the EM model. Solving the EM model in general is, however, a very hard problem. We report in this paper several partial results on the asymptotic solutions for statistical programs. While our results do not add up to a comprehensive solution to statistical programs they do offer explicit solutions in several cases, and shed light on the nature of the EM model for database statistics.

1.2   Main Technical Results

In this paper, we introduce classify programs according to two axes. First, whether the statistical assertions are on base tables only, or on both base tables and views: we say that Σ is in *normal form* (NF) if all statistical assertions are on base tables; otherwise it is in non-NF. The program in Fig 1 is in non-NF. Second, whether the views in Σ, Γ have joins or are join-free: we call the program *composite* if all views have joins, and *atomic* if all are join-free (we do not consider mixed atomic/composite programs). All four combinations are possible, e.g., an NF, composite program means that all statistical assertions are on base tables and the inclusion constraints are composite views. In this paper, we consider only *composite programs*[3].

---

[3] It turns out that atomic and mixed programs require an entirely different set of more complex, analytic techniques. In the full version of this paper [13], we discuss our

We give a complete solution for composite, NF programs: given statistics on the base tables and a set of full inclusion constraints, the EM model is described by an explicit formula. We prove this by using techniques from [6]. Second, we prove a more limited result for non-NF programs, by giving an explicit formula when the views are restricted to *project-semi-joins*. This explicit formula gives an important insight into the nature of difficulty of the non-NF programs, as we explain below.

In addition to these explicit solutions, we discuss a generic technique that we encapsulate as the *conditioning theorem* (Sec. 4.1); this reduces a more complex program to a simpler program plus additional inclusion constraints; this is what has motivated us to study statistical programs *together* with constraints.

The conditioning theorem states that every solution $\mathbf{P}$ to the EM model can be expressed as a conditional probability $\mathbf{P}(-) = \mathbf{P}_0(- \mid \Gamma)$, where $\mathbf{P}_0$ is a tuple-independent distribution called the *prior*, and Γ is a set of inclusion and set-equality constraints. Since $\mathbf{P}_0$ is tuple-independent, it is specified by a number $p_i \in (0, 1)$, one for each relation R_i, representing the probability that a generic tuple in the domain belongs to R_i; the corresponding *odds* is denoted $\alpha_i = p_i/(1 - p_i)$. Understanding the quantity $\mathbf{P}_0(\Gamma)$ is a key component to understanding the EM model. This quantity converges to 1 for composite NF programs, and to 0 for composite NNF programs, suggesting that the techniques used to solve composite, NF programs (which turn out to be simpler) do not extend to the other cases.

The EM model computation that we study in this paper is the first step in our program of using the EM framework for query size estimation. The second step is to use the model in order to do query size estimation, which we leave for future work, noting that it has been solved for the special case of independent distributions [6].

The rest of the paper is organized as follows. We describe the architecture of our proposed optimizer in Sec. 2, the EM model in Sec. 3, discuss composite programs in Sec. 4, and discuss how to handle range predicates in Sec. 5. We conclude in Sec. 7.

2 Cardinality Estimation Architecture Under EM

In this section, we describe in detail the architecture of a cardinality estimation that we envision. Figure 2 shows the architecture of a Volcano style query optimizer. The cardinality estimation module is shaded since this is our focus. We leverage execution feedback similar to prior work. As queries execute in the system, the cardinalities of various expressions are monitored and stored in a feedback cache. This cache constitutes additional information for the cardinality estimation module to use.

In today's query optimizer, synopses such as histograms are used for cardinality estimation. They are built over the underlying database. Recent work addresses the problem of refining these synopses using execution feedback. However,

---

preliminary results for atomic programs. We show for example that *any* program can be transformed to a normal form program. Interestingly, this normalization process also introduces additional inclusion constraints.
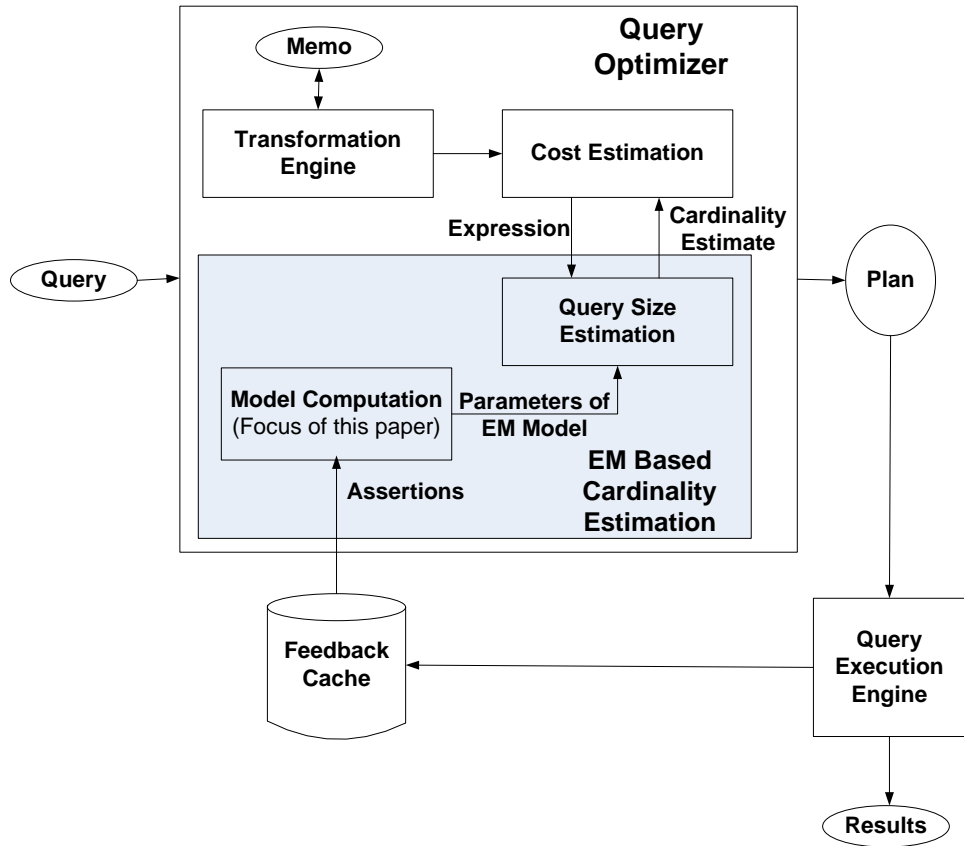
Fig. 2. EM based Cardinality Estimation

these synopses are restricted in that they only cater to a small class of queries. For example, single-table histograms only cater to point and range queries over the same table. In order to generate estimates for complex queries, assumptions such as the independence assumption and containment assumption are used [10] which are known to yield significant estimation errors. Further, it is far from clear how the execution feedback for such complex expressions which could counter the estimate yielded by using the synopsis structures can be used to refine the synopses. For instance, it is not clear how feedback about join cardinalities can be used to refine single-dimensional histograms.

Instead, in this paper we consider a fundamentally different approach to cardinality estimation. Here, the only input to the cardinality estimation module is assertions about the database which consist of query expressions and their true

cardinality. These assertions could be obtained directly from the data itself or from execution feedback. The goal of the cardinality estimation module then is to use these assertions to estimate the cardinality of a new query expression.

Since the input assertions could in principle be consistent with multiple database instances, they are interpreted as constraints and used to impose a distribution on this space of possible worlds. The cardinality estimation then finds the expected size of the new expression over this distribution.

We use the entropy maximization principle to identify the distribution. Thus, the cardinality estimation module consists of a *model computation* component which uses the input assertions to find parameters that encode the EM distribution. These parameters are used by a *query size estimation* component to compute the expected size of an expression over this distribution.

3   The EM Model

We introduce basic notations then review the EM model. `CQ` denotes the class of conjunctive queries over a relational schema $R_1, \ldots, R_m$. We define a *project-join* query as a conjunctive query without constants and where no subgoal has repeated variables, and write `PJ` for the class of project-join queries. For example $R(x, y), S(y, z, u)$ is a project-join query, but neither $R(a, x)$ nor $S(x, x, y), T(y, z)$ are. An arithmetic predicate, or range predicate, has the form x `op` c, where `op` $\in \{<, \leq, >, \geq\}$ and c is a constant; we denote by `PJ`$^{\leq}$ the set of project-join queries with range predicates.

Let Γ be a set of *full inclusion constraints*, i.e., statements of the form $\forall \bar{x}.w(\bar{x}) \Rightarrow R_i(\bar{x})$, where $w \in$ `PJ`$^{\leq}$ and R_i is a relation name.

3.1   Background: The EM Model

For a fixed domain D and constraints Γ we denote $\mathcal{I}(\Gamma)$ the set of all instances over D that satisfy Γ; the set of all instances over D is $\mathcal{I}(\emptyset)$, which we abbreviate $\mathcal{I}$. A probability distribution on $\mathcal{I}(\Gamma)$ is a set of numbers $\bar{p} = (p_I)_{I \in \mathcal{I}(\Gamma)}$ in $[0, 1]$ that sum up to 1. We use the notations p_I and $\mathbf{P}[I]$ interchangeably in this paper.

A *statistical program* is a pair $\Sigma = (\bar{v}, \bar{d})$, where $\bar{v} = (v_1, \ldots, v_s)$ are project-join queries, $v_i \in$ `PJ`, and $(d_1, \ldots, d_s)$ are positive real numbers. A pair (v_i, d_i) is a *statistical assertion* that we write informally as $\#v_i = d_i$; in the simplest case it can just assert the cardinality of a relation, $\#R_i = d_i$. A probability distribution $\mathcal{I}(\Gamma)$ *satisfies* a statistical program Σ if $E[|v_i|] = d_i$, for all $i = 1, m$. Here $E[|v_i|]$ denotes the expected value of the size of the view v_i, i.e., $\sum_{I \in \mathcal{I}} |v_i(I)| p_I$. We will also allow the domain size N to grow to infinity. For fixed values $\bar{d}$ we say that a sequence of probability distributions $(\bar{p}_N)_{N>0}$ *satisfies* $\Sigma = (\bar{v}, \bar{d})$ *asymptotically* if $\lim_{N \to \infty} E_N[|v_i|] = d_i$, for $i = 1, m$.

Given a program Σ, we want to determine the most "natural" probability distribution $\bar{p}$ that satisfies Σ and use it to estimate query cardinalities. In general, there may not exist any probability distribution that satisfies Σ; in this case, we say that Σ is unsatisfiable. On the other hand, there may exist many solutions. To choose a canonical one, we apply the Entropy Maximization (EM) principle.

Definition 1. *A probability distribution* $\bar{p} = (p_I)_{I \in \mathcal{I}(\Gamma)}$ *is an EM distribution associated to* Σ *if the following two conditions hold: (1)* $\bar{p}$ *satisfies* Σ, *and (2) it has the maximum entropy among all distributions that satisfy* Σ, *where the entropy is* $H = -\sum_{I \in \mathcal{I}(\Gamma)} p_I \log p_I$.

With slight abuse, we refer to an EM distribution as *the* EM model, assuming it is unique. For a simple illustration, consider the following program on the relation $R(A, B, C)$: $\#R = 200$, $\#R.A = 40$, $\#R.B = 30$, $\#R.C = 20$. Thus, we know the cardinality of R, and the number of distinct values of each of the attributes A, B, C. We want to estimate $\#R.AB$, i.e., the number of distinct values of pairs AB. Clearly this number can be anywhere between 40 and 200, but currently there does not exists a principled approach for query optimizers to estimate the number of distinct pairs AB from the other four statistics. The EM model gives such a principled approach. According to this model, R is a random instance over a large domain D of size N, according to a probability distribution described by the probabilities p_I, for $I \subseteq D^3$. The distribution p_I is defined precisely: it satisfies the four statistical assertions above, and is such that the entropy is maximized. Therefore, the estimate we seek also has a well defined semantics, as $E[\#R.AB] = \sum_{I \subseteq D^3} p_I |I.AB|$. This estimate will certainly be between 40 and 200; it will depend on N, which is an undesirable property, but a sensible thing to do is to let N grow to infinity, and compute the limit of $E[\#R.AB]$. Thus, the EM model offers a principled and uniform approach to query size estimation. Of course, in order to compute any estimate we must first find the EM distribution p_I; this is the goal in this paper.

To describe the general form of an EM distribution, we need some definitions. Fix the set of constraints Γ and the views $\bar{v} = (v_1, \ldots, v_s)$.

Definition 2. *The* partition function *for* Γ *and* $\bar{v}$ *is the following polynomial* T *with* s *variables* $\bar{x} = (x_1, \ldots, x_s)$:

$$T^{\Gamma, \bar{v}}(\bar{x}) = \sum_{I \in \mathcal{I}(\Gamma)} x_1^{|v_1(I)|} \cdots x_s^{|v_s(I)|}$$

Let $\bar{\alpha} = (\alpha_1, \ldots, \alpha_s)$ *be* s *positive real numbers. The probability distribution associated to* $(\Gamma, \bar{v}, \bar{\alpha})$ *is:*

$$p_I = \omega \alpha_1^{|v_1(I)|} \cdots \alpha_s^{|v_s(I)|} \tag{1}$$

where $\omega = 1/T^{\Gamma, \bar{v}}(\bar{\alpha})$.

We write T instead of $T^{\Gamma, \bar{v}}$ when $\Gamma, \bar{v}$ are clear from the context. The partition function can be written more compactly as:

$$T(\bar{x}) = \sum_{k_1, \ldots, k_s} C_\Gamma(N, k_1, \ldots, k_s) x_1^{k_1} \cdots x_s^{k_s}$$

where $C_\Gamma(N, k_1, \ldots, k_s)$ denotes the number of instances I over a domain of size N that satisfy Γ and for which $|v_i(I)| = k_i$, for all $i = 1, s$.

The following is a key characterization of EM distributions.

Theorem 1. *[12, page 355] Let $\Sigma = (\bar{v}, \bar{d})$ be a statistical program. For any probability distribution $\bar{p}$ that satisfies the statistics Σ the following holds: $\bar{p}$ is an EM distribution iff there exist parameters $\bar{\alpha}$ s.t. $\bar{p}$ is given by the Equation (1) (equivalently: $\bar{p}$ is associated to $(\Gamma, \bar{v}, \bar{\alpha})$).*

The message of this theorem is that the weight of an instance I under the EM distribution *only depends on $|v_i(I)|$*. That is, the distribution depends exactly on the provided statistics and makes no additional assumptions. It is this property that makes the EM distribution the natural model for database statistics. In a Bayesian sense, for a fixed set of statistics the EM model yields *the optimal estimate*. We refer to Jaynes [12, page 355] for a full proof and further discussion of this point; the "only if" part of the proof is both simple and enlightening, and we include in the Appendix for completeness.

We illustrate the utility of this theorem with two simple examples:

Example 2. **The Binomial-Model** Consider a relation $R(A, B)$ and the statistical assertion $\#R = d$ with $\Gamma = \emptyset$. The partition function is the binomial, $T(x) = \sum_{k=0,N^2} \binom{N^2}{k} x^k = (1 + x)^{N^2}$ and the EM model turns out to be the probability model that randomly inserts each tuple in R independently, with probability $p = d/N^2$. We need to check that this is an EM distribution: given an instance I of size k, $\mathbf{P}[I] = p^k(1 - p)^{N^2 - k}$, which we rewrite as $\mathbf{P}[I] = \omega \alpha^k$. Here $\alpha = p/(1 - p)$ is the *odds* of a tuple, and $\omega = (1 - p)^{N^2} = \mathbf{P}[I = \emptyset]$. This is indeed an EM distribution by Theorem 1. Asymptotic query evaluation on a generalization of this distribution to multiple tables was studied in [5]. □

Example 3. **Overlapping Ranges** Consider two views[4]:

$$v_1(x, y) \;\text{:-}\; R(x, y), x < .60N \text{ and } v_2(x, y) \;\text{:-}\; R(x, y), .25N \leq x$$

and the statistical program $\#v_1 = d_1$, $\#v_2 = d_2$ (again $\Gamma = \emptyset$). Assuming $N = 100$, the views partition the domain into three buckets, $D_1 = [1, 24]$, $D_2 = [25, 59]$, $D_3 = [60, 100]$, of sizes N_1, N_2, N_3. Here we want to say that we observe d_1 tuples in $D_1 \cup D_2$ and d_2 tuples in $D_2 \cup D_3$. The EM model gives us a precise distribution that represents only these observations and nothing more. The partition function is $(1 + x_1)^{N_1}(1 + x_1 x_2)^{N_2}(1 + x_2)^{N_3}$, and the EM distribution has the form $\mathbf{P}[I] = \omega \alpha_1^{k_1} \alpha_2^{k_2}$, where $k_1 = |I \cap (D_1 \cup D_2)|$ and $k_2 = |I \cap (D_2 \cup D_3)|$; we show in Sec. 5 how to compute the parameters α_1, α_2.

In this paper we study the *model computation problem*: given a statistical program Σ, find the parameters $\bar{\alpha}$ for the EM model. The ultimate goal of our program is to further use these parameters to estimate the size of arbitrary queries, but we will not treat the latter problem in this paper. The model depends on the size of the domain, N, and this is an undesirable property, since in practice N has no meaning other than that it is large. For that reason, we study the *asymptotic model computation problem* in this paper: find a sequence of parameters $\bar{\alpha}_N$ s.t. the distribution associated to $(\Gamma, \bar{v}, \bar{\alpha}_N)$ satisfies Σ asymptotically.

---

[4] We represent range predicates as fractions of N so we can allow N to go to infinity.

To simplify our discussion we present our results for the case when the queries in the statistical program have no range predicates, and show in Sec. 5 how to handle range predicates. Thus, from now on, until Sec. 5, we will assume all conjunctive queries to be without range predicates.

3.2 A Taxonomy for Statistical Programs

Recall that PJ denotes the class of project-join queries. We define here two sub-classes. First, a *project query* is a single subgoal query without constants or repeated variables; denote P the class of project queries. Second, a *single component join query* is a project-join query with the following properties: it is minimized, has at least two subgoals, and has a single connected component; denote PJ^C the class of single component join queries. Queries V_3, V_4 in Fig. 1 are in P; queries V_5 is in PJ^C. P and PJ^C are two disjoint subclasses of PJ that do not cover PJ. Some queries in PJ are not in either class, e.g. $v(x, y)$:- $R(x, y), R(z, y), S(u)$ is a query that minimizes to $R(x, y), S(u)$, which is neither in P nor in PJ^C (it has two connected components): we do not treat such queries in this paper.

We classify statistical programs $\Sigma = (\bar{v}, \bar{d})$ and constraints Γ along two axes:

Definition 3. Σ *is in* normal form *(NF) if all statistical assertions are on base tables; otherwise, it is in* non-normal form *(NNF).*

Definition 4. *(1)* Σ *is* composite *if for every statistical assertion* $\#v = d$, v *is either a base table or is in* PJ^C. Γ *is* composite *if for every constraint* $\forall \bar{x}.w(\bar{x}) \Rightarrow R_i(\bar{x})$, w *is in* PJ^C. *We say that* Σ, Γ *is* composite *if both are composite. (2)* Σ, Γ *is* atomic *if all their views are in* $\mathcal{P}$.

Thus, there are four combinations of programs: NF/NNF and composite/atomic. For example, referring to Fig. 1, the program (Σ_1, Γ_1) $\Sigma_1 = \{\#R = 3000, \#T = 42000\}$ with $\Gamma_1 = \{\gamma_1\}$ is an atomic, NF program; if we add the statistical assertion (V_4, d_4), then the program is still atomic, but no longer in normal form. On the other hand, (Σ_1, Γ) with $\Gamma = \{\gamma_3\}$ is composite and in normal form; if we add the statistic (V_5, d_5) then this becomes a composite program, not in normal form. We do not treat mixed atomic/composite programs.

4 Composite Programs

We start by discussing the case when all queries are *composite*. First, we introduce the two main techniques used in this section, conditioning on the prior, and the asymptotic probabilities from [5], then we give our results.

4.1 From Conditionals to EM Models

Recall that $\mathcal{I}$ denotes the set of all database instances, without any constraints. Define a *prior probability distribution* to be any tuple-independent probability distribution $\mathbf{P}_0$ on $\mathcal{I}$. As seen in Example 2, this is an EM distribution for a very simple NF program, which just asserts the cardinalities of each relation, $\#R_i = d_i$, and has no constraints. Each tuple t into R_i has probability $\mathbf{P}_0[t] =$

$d_i/N^{arity(R_i)}$, and the EM parameters are $\alpha_i = p_i/(1-p_i) \approx p_i$. Now let's add a set of constraints Σ, i.e., consider the NF program consisting both of cardinality assertions $\#R_i$ and constraints Σ. Its EM model is obtained as follows:

Theorem 2 (Conditioning). *Let $\mathbf{P}$ be the EM model for a NF program Σ, Γ. Then there exists a prior probability distribution $\mathbf{P}_0$ such that:*

$$\forall\, I \in \mathcal{I}(\Gamma), \;\; \mathbf{P}[I] = \mathbf{P}_0[I \mid \Gamma]$$

Moreover, the expected values are obtained through the following transfer equation: $E[|q|] = E_0[|q| \mid \Gamma]$.

Proof. Let $\bar{\alpha}$ be the parameters of $\mathbf{P}$. Define the tuple-independent prior as follows: for each relation R_i, define $\mathbf{P}_0[t \in R_i] = p_i = \alpha_i/(1 + \alpha_i)$. (Thus, the odds of p_i are precisely α_i.) Then $\mathbf{P}_0[I] = \omega_0 \prod \alpha_i^{|R_i^I|}$ (follows by generalizing Example 2) and $\mathbf{P}[I] = \omega \prod \alpha_i^{|R_i^I|}$ (by definition). Thus, $\mathbf{P}$ and $\mathbf{P}_0$ are essentially the same expression, only $\mathbf{P}$ is defined over a restricted domain $\mathcal{I}(\Gamma) \subseteq \mathcal{I}$.

For a simple illustration, consider the statistical program Σ: $\#R(A,B) = d_1$, $\#T(B,C) = d_2$, and the constraints Γ: $R(x,y), R(y,z) \Rightarrow R(x,z)$ and $T(x,y), R(y,z) \Rightarrow T(z,x)$. To solve it, first solve a different, simpler program $\#R(A,B) = b_1$, $\#T(B,C) = b_2$, without constraints. This is a tuple-independent probability distribution $\mathbf{P}_0$. Then the solution to Σ, Γ is given as $\mathbf{P}[I] = \mathbf{P}_0[I \mid \Gamma]$. The difficulty lies in choosing the statistics b_1, b_2 of the simpler model: we need to ensure that $E_0[|R| \mid \Gamma] = d_1$, $E_0[|T| \mid \Gamma] = d_2$.

4.2 Background: Asymptotic Query Probabilities

Based on our discussion, we need to study prior probabilities that have the form $\mathbf{P}[t \in R] = b(R)/N^{arity(R_i)}$, where $b(R)$ is a constant that depends only on the relation symbol R. These tuple-independent distributions were studied in [5]. It was shown that for any Boolean conjunctive query $q \in CQ$, there exists two constants $E(q)$ and $C(q)$, which can be computed only from the constants $b(R)$ and the query expression, s.t. $\mathbf{P}[q] = C(q)/N^{E(q)} + O(1/N^{E(q)+1})$. We give the expressions for $C(q)$ and $E(q)$ in Fig. 3.

Example 4. We illustrate the notations in Fig. 3 on the query $q = R(x,y), R(y,z)$, and $b(R) = b$. $D(q) = 4 - 3 = 1$ and is called the degree of q; and $b(q) = b^2$. $UQ(q)$ is obtained by substituting variables in q and contains four queries (up to isomorphism): q itself, then $R(x,x), R(x,y)$, then $R(x,y), R(y,y)$, and finally $R(x,x)$. Their degrees are $1, 2, 2, 1$ respectively, thus $E(q) = 1$ and is called the exponent of q. $UQ^0(q)$ consists of the first and last queries (those that have $D = 1$), and $aut(q_0)$ is the number of automorphisms for q_0, and is 1 for both queries in $UQ^0(q)$. Finally, $C(q) = b^2 + b$ is called the *coefficient* of q. Thus, $\mathbf{P}[q] = (b^2 + b)/N + O(1/N^2)$.

We consider here only conjunctive queries where all connected components have $E > 0$; this rules out some degenerate queries, whose treatment is more complex [4]. All PJ^C queries satisfy this property, since they have a single component and $E > 0$.

$$V(q) = \text{the number of distinct variables in } q$$
$$a(q) = \sum \{arity(g) \mid g \in goals(q)\}$$
$$D(q) = a(q) - V(q)$$
$$b(q) = \prod \{b(g) \mid g \in goals(q)\}$$
$$UQ(q) = \{\eta(q) \mid \eta = \text{a substitution of variables}\}$$
$$E(q) = \min D(q_0) q_0 \in UQ(q)$$
$$UQ^0(q) = \{q_0 \mid q_0 \in UQ(q), D(q_0) = E(q)\}$$
$$C(q) = \sum_{q_0 \in UQ^0(q)} \frac{b(q_0)}{aut(q_0)}$$

Fig. 3. Notations for Theorem 3 from [5].

Theorem 3. *[5] For any conjunctive query $q \in CQ$, $\mathbf{P}_0(q) = C(q)/N^{E(q)} + O(1/N^{E(q)+1})$.*

4.3 Composite NF Programs

Theorem 4 (Composite, NF). *Consider a statistical program in normal form Σ: $\#R_j = d_j$, for $j = 1, m$. Consider a set of inclusion constraints Γ where all queries are composite. Then an asymptotic solution to the EM model is given by $\alpha_j = d_j/N^{arity(R_j)}$.*

The proof uses Theorem 3 and is given in the appendix; it uses Theorem 3 as well as specific properties of the expressions D and E in Fig. 3. At a high level, the proof exploits the fact that $\lim_N \mathbf{P}_0[\Gamma] = 1$ (i.e., the constraints, Γ, almost surely hold), where $\mathbf{P}_0$ is the prior associated to the same statistical program (Σ, Γ): that is, the constraints Γ holds almost certainly in the prior, and hence the statistics are not affected by conditioning.

Example 5. Consider the constraints $R(x, y), R(y, z) \Rightarrow R(x, z)$, and $T(x, y, z), R(y, u) \Rightarrow S(y)$, and the statistical assertions $|R| = d_1$, $|T| = d_2$, $|S| = d_3$. An asymptotic solution to the EM model is given by $\alpha_1 = d_1/N^2$, $\alpha_2 = d_2/N^3$, $\alpha_3 = d_3/N$.

4.4 Composite Non-NF Programs

Let Σ be a statistical program that consists of assertions on all relations, $\#R_j = d_j$, as well as assertions over composite views, $\#q_i = d_i$. Create a new relation symbol T_i for each statistical assertion of the same arity as the view, and define the *set equality constraints* $\forall \bar{x}.(\exists \bar{y}.q_i(\bar{x}, \bar{y}) \iff T_i(\bar{x}))$. Each set equality constraint is expressed as $\gamma_i \wedge \delta_i$, where γ_i is a full inclusion constraint and δ_i is a reverse inclusion constraint:

$$\gamma_i \equiv \forall \bar{x}.q_i(\bar{x}) \Rightarrow T_i(\bar{x})$$
$$\delta_i \equiv \forall \bar{x}.T_i(\bar{x}) \Rightarrow (\exists \bar{y}.q_i(\bar{x}, \bar{y}))$$

Denote Δ and Γ the set of all inclusion- and all reverse inclusion constraints. As before, the EM solution is given by a conditional $\mathbf{P}[I] = \mathbf{P}_0[I \mid \Delta \wedge \Gamma]$, where $\mathbf{P}_0$ is some tuple-independent prior. However, it is now more difficult to transfer the expected sizes, and we provide a closed form solution only for a restricted class of views.

Definition 5. *A query $q \in PJ$ is a* project-semi-join *query if the following conditions hold. Let $\bar{x} = (x_1, \ldots, x_k)$ be its head variables:*

- *q has no self-joins (i.e., no repeated relation symbol).*
- *If two different subgoals in q share a variable y, then $y \in \bar{x}$.*
- *For every subgoal g, if g contains x_i then it also contains x_{i+1}.*

A *core subgoal* is a subgoal that contains the smallest number of head variables. The core of q is the set of core subgoals and denoted G. In what follows, a transfer equation is an equation that relates a size estimate under the prior distribution to the size estimate of the distribution under constraints.

Lemma 1. *Let δ be an inverse inclusion constraint $\forall \bar{x}.T(\bar{x}) \Rightarrow \exists \bar{y}.q(\bar{x}, \bar{y})$ where q is a project-semi-join query. Define the prior:*

$$\mathbf{P}_0(t \in R_j) = \frac{b(R_j)}{N^{arity(R_j)}}$$

$$\mathbf{P}_0(t \in T) = 1 - \frac{b(T)}{N^{E(G)}}$$

(Here $E(G)$ denotes the exponent of the core, see Fig. 3.) Let $R_1, \ldots, R_m$ be all subgoals in q, $m \geq 1$. Then the transfer equation for the view is:

$$E_0[|T| \mid \delta] = \frac{\prod_{R \in goals(q)} b(R)}{b(T)} \tag{2}$$

The transfer equation for any other relation in R is as follows. If the query consists only of the core, then:

$$E_0[|R_i| \mid \delta] = b(R_i) + \frac{C(G)}{b(T)} \tag{3}$$

($C(G)$ is the coefficient of G, see Fig. 3.) If the query has subgoals other than the core, then the expected cardinalities are unchanged.

We prove the lemma in the appendix. From the lemma we derive:

Theorem 5 (Composite, non-NF). *Let Σ be a statistical program where all queries are project-semi-join queries, and do not share common subgoals. Then an EM model for Σ has the following parameters:*

- *For every base relation R_j, the parameter is $\alpha_j = b_j / N^{arity(R_j)}$.*
- *For every view assertion v_j, the parameter is $\alpha_j = N^{E(G)} / b_j$, where G is the core of v_j.*

where the numerical values b_j are obtained by solving a system of equations (2) and (3).

It is interesting to compare the solution to an NF program to that of a non-NF program (Theorems 4 and 5). For NF programs all parameters have the form $\alpha_i = d/N^a$, for integer $a > 0$. For non-NF programs some parameters have the form N^a/d and, thus, go to infinity.

Example 6. Consider the following statistical program[5]:

$$\#R_1 = d_1 \quad \#R_2 = d_2$$
$$v(x) \;\; \text{:-} \;\; R_1(x), R_2(x) \quad \#v = d_3$$

Thus, we are given the sizes of R_1, R_2, and of their intersection. We introduce a new relation symbol T and the constraint $\delta = T(x) \Leftrightarrow R_1(x), R_2(x)$, then define the program in normal form:

$$\#R_1 = d_1 \quad \#R_2 = d_3 \quad \#T = d_3$$

The theorem gives us the EM solution as follows. The core is the entire query, hence we define the prior:

$$\mathbf{P}_0(R_1(a)) = e_1/N \quad \mathbf{P}_0(R_2(a)) = e_2/N \quad \mathbf{P}_0(T(a)) = 1 - e_3/N$$

where $\mathbf{P}_0(R_1(a))$ denotes the marginal probability of the tuple $R_1(a)$. Note that T has a very large probability. This gives us an EM model to our initial statistical program if we solve e_1, e_2, e_3 in:

$$d_3 = \frac{e_1 e_2}{e_3}$$
$$d_1 = e_1 + \frac{e_1 e_2}{e_3}$$
$$d_3 = e_2 + \frac{e_1 e_2}{e_3}$$

Example 7. A more complex example is a statistical program that uses the following project-semi-join view:

$$v_6(x_1, x_2, x_3) \;\; \text{:-} \;\; R_1(x_1, x_2, x_3, y), R_2(x_2, x_3), R_3(x_2, x_3, z),$$
$$R_4(x_1, x_2, x_3), R_5(x_1, x_2, x_3)$$

The core consists of R_2 and R_3, and so we define $\mathbf{P}_0[t \in T] = e_T/N^2$, where e_T is chosen such that $d_2 d_3/e_T = d_6$.

---

[5] Its partition function is $(1 + x_1 + x_2 + x_1 x_2)^N$. Intuitevely, this is because each of the n tuples is in R_1 and so pays x_1, is in R_2 and so pays x_2 or is in both R_1 and R_2 and so pays $x_1 x_2$.

5 Bucketization

Finally, we re-introduce range predicates like $x < c$, both in the constraints and in the statistical assertions. To extend the asymptotic analysis, we assume that all constants are expressed as fractions of the domain size N, e.g., in Ex. 3 we have $v_1(x,y)$:- $R(x,y), x < 0.25N$.

Let $\bar{R} = R_1, \ldots, R_m$ be a relational schema, and consider a statistical program Σ, Γ with range queries, over the schema $\bar{R}$. We translate it into a *bucketized* statistical program Σ^0, Γ^0, over a new schema $\bar{R}^0$, as follows. First, use all the constants that occur in the constraints or in the statistical assertions to partition the domain into b buckets, $D = D_1 \cup D_2 \cup \ldots \cup D_b$. Then define as follows:

- For each relation name R_j of arity a define b^a new relation symbols, $R_j^{i_1 \cdots i_a} = R_j^{\bar{i}}$, where $i_1, \ldots, i_a \in [b]$; then $\bar{R}^0$ is the schema consisting of all relation names $R_j^{i_1 \cdots i_a}$.
- For each conjunctive query q with range predicates, denote $\texttt{buckets}(q) = \{q^{\bar{i}} \mid \bar{i} \in [b]^{|Vars(q)|}\}$ the set of queries obtained by associating each variable in q to a unique bucket, and annotating the relations accordingly. Each query in $\texttt{buckets}(q)$ is a conjunctive query over the schema $\bar{R}^0$, without range predicates, and q is logically equivalent to their union.
- Let $BV = \bigcup\{\texttt{buckets}(v) \mid (v,d) \in \Sigma\}$ (we include in BV queries up to logical equivalence), and let c_u denote a constant for each $u \in BV$, s.t. for each statistical assertion $\#v = d$ in Σ the following holds

$$\sum_{u \in \texttt{buckets}(v)} c_u = d \tag{4}$$

Denote Σ^0 the set of statistical assertions $\#u = c_u$, $u \in BV$.
- For each inclusion constraint $w \Rightarrow R$ in Γ, create $b^{|Vars(w)|}$ new inclusion constraints, of the form $w^{\bar{j}} \Rightarrow R^{\bar{i}}$; call Γ^0 the set of new inclusion constraints.

Then the following holds:

Proposition 1. *Let Σ^0, Γ^0 be the bucketized program for Σ, Γ. Let $\bar{\beta} = (\beta_k)$ be the EM model of the bucketized program. Consider some parameters $\bar{\alpha} = (\alpha_j)$. Suppose that for every statistical assertion $\#v_j = d_j$ in Σ condition (4) holds, and the following condition holds for every query $u_k \in BV$:*

$$\beta_k = \prod_{j : u_k \in \texttt{buckets}(v_j)} \alpha_j \tag{5}$$

Then $\bar{\alpha}$ is a solution to the EM model for Σ, Γ.

This gives us a general procedure for solving the EM model for programs with range predicates: introduce new unknowns $c_j^{\bar{i}}$ and add Equations (4) and (5), then solve the EM model for the bucketized program under these new constraints.

Example 8. Recall Example 3: we have two statistics $\#\sigma_{A \leq 0.60N}(R) = d_1$, and $\#\sigma_{A \geq 0.25N}(R) = d_2$. The domain D is partitioned into three domains, $D_1 = [1, 0.25N)$, $D_2 = [0.25N, 0.60N)$, and $D_3 = [0.60N, N]$, and we denote N_1, N_2, N_3 their sizes. The bucketization procedure is this. Define a new schema R^1, R^2, R^3, with the statistics $\#R^1 = c^1$, $\#R^2 = c^2$, $\#R^3 = c^3$, then solve it, subject to the Equations (5):

$$\beta_1 = \alpha_1$$
$$\beta_2 = \alpha_1\alpha_2$$
$$\beta_3 = \alpha_2$$

We can solve for R^1, R^2, R^3, since each R^i is given by a binomial distribution with tuple probability $\beta_i/(1 + \beta_i) = c^i/N_i$. Now use Equations (4), $c^1 + c^2 = d_1$ and $c^2 + c^3 = d_2$ to obtain:

$$N_1 \frac{\alpha_1}{1 + \alpha_1} + N_2 \frac{\alpha_1\alpha_2}{1 + \alpha_1\alpha_2} = d_1$$
$$N_3 \frac{\alpha_2}{1 + \alpha_2} + N_2 \frac{\alpha_1\alpha_2}{1 + \alpha_1\alpha_2} = d_2$$

Solving this gives us the EM model. Consistent histograms [16] had a similar goal of using EM to capture statistics on overlapping intervals, but use a different, simpler probabilistic model based on frequencies.

6 Related Work

There are two bodies of work that are most closely related to this paper. The first consists of the work in cardinality estimation. As noted above, while a variety of synopses structures have been proposed for cardinality estimation [1, 8, 10, 15], they have all focused on various sub-classes of queries and deriving estimates for arbitrary query expressions has involved *ad-hoc* steps such as the independence and containment assumptions which result in large estimation errors [11]. In contrast, we ask the question what is the framework for performing cardinality estimation over arbitrary expressions in the presence of incomplete information. We approach this task via the EM principle.

The EM model has been applied in prior work to the problem of cardinality estimation [14, 16]. However, the focus was restricted to queries that consist of conjunctive selection predicates over single tables. In contrast, we explore a full-fledged EM model that can incorporate statistics involving arbitrary first-order expressions.

Another body of related work consists of the work in probabilistic databases [7] which focuses on efficient query evaluation over a probabilistic database. The input statistics impose many possible distributions over the *possible worlds* and we choose the distribution that has maximum entropy. Our focus in this paper is in deriving the parameters of this EM distribution. The related problem of query estimation for a given model is not addressed in this paper. This is closely related to the problem of evaluating queries over probabilistic databases.

Finally, we observe that entropy-maximization is a well-established principle in statistics for handling incomplete information [12]. As with probabilistic databases, new challenges emerge in the context of database systems, in our case the nature of statistics.

7   Conclusion

In this paper we propose to model arbitrary database statistics using an Entropy-Maximization probability distribution. This model is attractive because any query has a well-defined size estimate, all statistics are treated as a whole rather than as individual synopses, and the model extends smoothly when new statistics are added. We reported in this paper several results that give explicit asymptotic solutions to statistical programs in several cases. As part of our technical development we described a technique encapsulated as the conditioning theorem (Theorem 2) that is of independent interest and are likely to be applicable to other statistical programs.

We are leaving for future work the second part: using an EM model to obtain query size estimates. This has been solved in the past only for the independent case [6].

References

1. N. Alon, P. B. Gibbons, Y. Matias, and M. Szegedy. Tracking Join and Self-Join Sizes in Limited Storage. In *PODS*, 1999.
2. F. Bacchus, A. Grove, J. Halpern, and D. Koller. From statistical knowledge bases to degrees of belief. *Artificial Intelligence*, 87(1-2):75–143, 1996.
3. S. Chaudhuri, V. R. Narasayya, and R. Ramamurthy. Diagnosing Estimation Errors in Page Counts Using Execution Feedback. In *ICDE*, 2008.
4. N. Dalvi. Query evaluation on a database given by a random graph. *Theory of Computing Systems*, 2009. to appear.
5. N. Dalvi, G. Miklau, and D. Suciu. Asymptotic conditional probabilities for conjunctive queries. In *ICDT*, 2005.
6. N. Dalvi and D. Suciu. Answering queries from statistics and probabilistic views. In *VLDB*, 2005.
7. N. Dalvi and D. Suciu. Management of probabilistic data: Foundations and challenges. In *PODS*, pages 1–12, Beijing, China, 2007. (invited talk).
8. A. Deligiannakis, M. N. Garofalakis, and N. Roussopoulos. Extended wavelets for multiple measures. *ACM Trans. Database Syst.*, 32(2), 2007.
9. P. Erdös and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kut. Int. Kozl.*, 5:17–61, 1960.
10. Y. E. Ioannidis. The History of Histograms. In *VLDB*, 2003.
11. Y. E. Ioannidis and S. Christodoulakis. On the propagation of errors in the size of join results. In *SIGMOD*, May 1991.
12. E.T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, UK, 2003.
13. R. Kaushik, C. Ré, and D. Suciu. General database statistics using entropy maximization: Full version. Technical Report #05-09-01, University of Washington, Seattle, Washington, May 2009.

14. V. Markl, N. Megiddo, et al. Consistently estimating the selectivity of conjuncts of predicates. In *VLDB*, 2005.
15. F. Olken. *Random Sampling from Databases*. PhD thesis, University of California at Berkeley, 1993.
16. U. Srivastava, P. Haas, V. Markl, M. Kutsch, and T. M. Tran. ISOMER: Consistent histogram construction using query feedback. In *ICDE*, 2006.
17. M. Stillger, G. M. Lohman, V. Markl, and M. Kandil. LEO - DB2's LEarning Optimizer. In *VLDB*, 2001.

A Proof of Theorem 1

The "only if" direction is very simple to derive by using the Lagrange multipliers for solving:

$$F_0 = \sum_{I \in \mathcal{I}} p_I - 1 = 0 \tag{6}$$

$$\forall i = 1, \ldots, s: \ F_i = \sum_{I \in \mathcal{I}} |v_i(I)| p_I - d_i = 0 \tag{7}$$

$$H = \text{maximum, where } \ H = \sum_{I \in \mathcal{I}} p_I \log p_I \tag{8}$$

According to that method, one has to introduce $s + 1$ additional unknowns, $\lambda, \lambda_1, \ldots, \lambda_s$: an EM distribution is a solution to a system of $|\mathcal{I}| + s + 1$ equations consisting of Eq.(6), (7), and the following $|\mathcal{I}|$ equations:

$$\forall I \in \mathcal{I}: \ \frac{\partial(H - \sum_{i=0,s} \lambda_i G_i)}{\partial p_I} = \log p_I - (\lambda_0 + \sum_{i=1,s} \lambda_i |v_i(I)|) = 0$$

This implies $p_I = \exp(\lambda_0 + \sum_{i=1,s} \lambda_i |v_i(I)|)$, and the claim follows by denoting $\omega = \exp(\lambda_0)$, and $\alpha_i = \exp(\lambda_i)$, $i = 1, s$.

B Proof of Theorem 4

A *type*, τ, of a set of variables $\bar{x}$ is a maximal consistent set of predicates of the form $x_i = x_j$ and $x_i \neq x_j$. Equivalently, a type is a partition of $\bar{x}$, and we denote $s(\tau)$ its size, i.e. the number of equivalence classes. For example, $x_1 = x_2 \neq x_3 = x_4 = x_5$ is a type of size 2. If q is a conjunctive query and $\bar{x}$ are some of its variables, then $q^\tau \in CQ(\neq)$ denotes the query obtained by adding the predicates τ to q. If $\bar{a}$ is a ground tuple of the same arity as $\bar{x}$, then the type of $\bar{a}$ is the unique type $\tau(\bar{a})$ s.t. $\bar{a}$ can substitute $\bar{x}^{\tau(\bar{a})}$. For example the type of (a, b, b, a) is $x_1 = x_4 \neq x_2 = x_3$. Recall that all results in [5] apply to queries in $CQ(\neq)$.

We start a few lemmas. Fix $q \in \text{PJ}$ a project-join quey. Recall that q has no constants. Let $\bar{x}$ be some of its variables, and let $\bar{a}$ be some constants of the same arity as $\bar{x}$.

Lemma 2. *If $q_0 \in UQ^0(q)$ then q_0 has no constants.*

Suppose $q_0 = \eta(q)$ had constants. Compute η in two steps: $q_1 = \eta_1(q)$ equates all variables as per the type of the constants, and $q_0 = \eta_0(q_1)$ maps some variables $\bar{x}$ one-to-one to constants. Then $D(q_0) = D(q_1) + |\bar{x}|$, contradicting the fact that q_0 has minimal degree.

Lemma 3. *Then $E(q[\bar{a}/\bar{x}]) = E(q^{\tau(\bar{a})}) + s(\tau(\bar{a})) \geq E(q) + s(\tau(\bar{a}))$.*

Continuing the notations above, for every $q_0 \in UQ(q[\bar{a}/\bar{x}])$ we have $D(q_0) = D(q_1) + |\bar{x}| = D(q_1) + s(\tau(\bar{a}))$, where $q_1 \in UQ(q^{\tau(\bar{a})})$. If we choose $q_0 \in UQ^0$ (i.e. minimize the value for $D(q_0)$) then we obtain the $\leq$ side of the inequality; if we choose $q_1 \in UQ^0$, then we minimize $D(q_1)$ and prove the other direction.

In other words, the lemma says that in order to compute the exponent of $q(\bar{a}/\bar{x})$ proceed as follows. First add in q all $=$ and $\neq$ predicates on $\bar{x}$ that are compatible with $\bar{a}$, call them τ, then compute $E[q^\tau]$, and finally add to this the number of distinct constants in $\bar{a}$.

Lemma 4. *Let $R(\bar{a})$ be a ground tuple for the relation R, and let $s = s(\tau(\bar{a}))$ be the size of its type (i.e. number of distinct constants in $\bar{a}$). Let $q \in PJ$ be a project-join query. Consider the conjuctive query $q, R(\bar{a})$, obtained by taking the conjunction of q and $R(\bar{a})$. Then (1) $E(q, R(\bar{a})) \geq E(q) + s$, and (2) there exists a tuple $\bar{a}$ s.t. $E(q, R(\bar{a})) = E(q) + s$.*

Let $\bar{z}$ be fresh variables. By the above lemma we have $E(q, R(\bar{a})) = E(q, R(\bar{z}^{\tau(\bar{a})})) + s(\tau(\bar{a}))$. Claim (1) follows from $E(q, R(\bar{z})) \geq E(q)$ and the previous lemma. For claim (2) there are two cases. If R does not appear in q, then $E(q, R(\bar{z}^\tau)) = E(q) + s(\tau)$ for any type τ (since R doesn't unify with anything in q), hence the claims holds for any choice of $\bar{a}$. If R appears in q, then $E(q, R(\bar{z})) = E(q)$. Indeed, consider any $q^0 \in UQ^0(q)$, and let τ be the type of any subgoal mentioning R. Then q^0 is also a homomrphic image of $q, R(\bar{z}^\tau))$, hence $E(q, R(\bar{z}^\tau)) \leq E(q^0) = E(q)$.

Proof. (of Theorem 4) Denote $\mathbf{P}_0$ the prior corresponding to the parameters α_j in the theorem. We start by computing $\mathbf{P}_0(\Gamma)$. Let $\gamma_1, \ldots, \gamma_p$ be the inclusion constraints in Γ. Each γ_k has the form $w_k \Rightarrow R_j$, hence its negation is $\neg\gamma_k \equiv w_k \wedge \neg R_j$. This is a conjunctive query with one negated subgoal. Using the inclusion/exclusion formula we obtain $\mathbf{P}_0[\Gamma] = 1 - \mathbf{P}_0[\neg\Gamma] = \sum_{S \subseteq [p]} (-1)^{|S|} \mathbf{P}_0[\wedge_{k \in S} \neg\gamma_k]$. For each subset $S \subseteq [p]$, denote Q_S the conjunctive query $Q_S \equiv \wedge_{k \in S} w_k$; by Theorem 3 we have $\mathbf{P}_0[Q_S] = O(1/N^{E(Q_S)})$, hence $\mathbf{P}_0[\wedge_{k \in S} \neg\gamma_k] \leq Pr[Q_S] = O(1/N^{E(Q_S)})$, which implies that $\mathbf{P}_0[\Gamma] = 1 - O(1/N)$.

Let R_j be a relation name and let t be a ground tuple in R_j. THen

$$E_0[R_j \mid \Gamma] = N^{arity(R_j)} \mathbf{P}_0[t \mid \Gamma]$$

On the other hand:

$$\mathbf{P}_0[t \mid \Gamma] = \frac{\mathbf{P}_0[\Gamma, t]}{\mathbf{P}_0[\Gamma]} = \frac{\mathbf{P}_0[t] - \mathbf{P}_0[t\neg\Gamma]}{1 - \mathbf{P}_0[\neg\Gamma]}$$

We have already seen that $\mathbf{P}_0[\neg \Gamma] = O(1/N)$. Similarly, $\mathbf{P}_0[t \neg \Gamma] = \mathbf{P}_0[t] * O(1/N)$. Indeed, for any conjuctive query Q_S defined above, $E(tQ_S) \geq E(Q_S) + s(\tau(t))$ hence $\mathbf{P}_0[tQ_s] = O(1/N^{E(tQ_S)}) \leq O(1/N^{s(\tau(t))+E(Q_S)})$ and the claim follows from the fact that $\mathbf{P}_0[t] = O(1/N^{s(\tau(t))})$.

We conclude that $\mathbf{P}_0[t \mid \Gamma] = \mathbf{P}_0[t]$, hence $E_0[R_j \mid \Gamma] = E[R_j]$.

C Proof of Theorem 4

Conjunctive queries are known to correspond to select-project-join query. Call a conjunctive query a *project-join* query if (a) there are no constants in its body, and (b) for any subgoal g occurring in q, all variable occurrences in g are distinct. For example, $R(x, y), R(y, z)$ is a project-join query, while $R(a, x)$ and $S(x, x, y)$ are not.

Lemma 5. *Let q be a project-join query with $E(q) > 0$. Let t be a ground tuple. Then $\lim_N \frac{\mathbf{P}(t,q)}{\mathbf{P}(t)} = 0$*

Proof. Note that both preconditions are necessary. If q has constants, then it fails: $t = R(a, b, c)$, $q = R(a, b, x)$, then $\mathbf{P}(tq) = O(1)/N^3$ because t and $R(a, b, x)$ unify to t hence $E(tq) = D(t) = 3$, and $\mathbf{P}(t) = O(1)/N^3$. If q has repeated variables, then it also fails: $t = R(a, a, a)$, $q = R(x, x, y), R(x, y, y)$, then $E(tq) = D(t) = 3$, hence both $\mathbf{P}(tq)$ and $\mathbf{P}(t)$ are $O(1)/N^3$. To see an example where it holds, consider $t = R(a, b, c)$, $q = R(x, y, z), S(z, u)$. To compute $E(tq)$ we compute $D(tq) = 8 - 4 = 4$, and $D(R(a, b, c), S(c, u)) = 5 - 1 = 4$. Hence $\mathbf{P}(tq) = O(1)/N^4$, while $\mathbf{P}(t) = O(1)/N^3$.

To prove the claim, we will show that $E(tq) > D(t)$. So, let $\eta(tq) \in UQ^0(tq)$. We have $\eta(tq) = t, \eta(q)$ (since η doesn't affect a ground tuple). If t doesn't appear as a subgoal in $\eta(q)$, then $E(tq) = D(t, \eta(q)) = D(t) + D(\eta(q)) \geq D(t) + E(q) > D(t)$. If t appears as a subgoal in $\eta(q)$, then it means that $\eta(tq) = \eta(q)$. In this case η maps at least one subgoal g in q to the tuple t. Consider the substitution η' that is like η, but whenever $\eta(x)$ is a constant, $\eta'(x)$ is set to a new distinguished variable. In other words, $\eta = \theta \circ \eta'$, where θ maps the new distinguished variables to the constants that they replaced. Note that $V(\eta(q)) \leq V(\eta'(q)) - A(t)$, because η must substitute at least the $A(t)$ distinct variables occurring in g with constants in order to equate it with t. Thus we have $E(tq) = D(\eta(q)) \geq D(t, \eta'(q))$ (because $A(\eta(q)) + A(t) = A(t\eta'(q))$), and the claim follows from $D(t, \eta'(q)) > D(t)$, which is similar to what we argued above.

Corollary 1. *Let γ be an inclusion constraint $\forall \bar{x}.(q \Rightarrow R)$, s.t. q is a project-join query with $E(q) > 0$. Let t be a ground tuple. Then $\lim_N \frac{\mathbf{P}(t, \neg \gamma)}{\mathbf{P}(t)} = 0$*

Proof. The negation is $\neg \gamma \equiv (q, \neg R)$, i.e. a conjunctive query with one negated subgoal. We have $\mathbf{P}(t, \neg \gamma) = \mathbf{P}(t, q, \neg R) \leq \mathbf{P}(t, q)$, and the claim follows from the previous lemma.

Corollary 2. *Let Γ be a set of inclusion constraints s.t. any left hand side of any constraint is a project-join query with $E > 0$. Let t be any ground tuple. Then $\lim_N \frac{\mathbf{P}(t|\Gamma)}{\mathbf{P}(t)} = 1$.*

Proof. By induction on the number of constraints in Γ. IF there is only one constraint γ, then $\mathbf{P}(t \mid \gamma) = \frac{\mathbf{P}(t\gamma)}{\mathbf{P}(\gamma)}$ and the claim follows from the previous corollary and the fact that $\mathbf{P}(\gamma) \to 1$. Consider a set of constraints of the form γ, Γ. Then $\mathbf{P}(t \mid \gamma, \Gamma) = \frac{\mathbf{P}(t,\gamma,\Gamma)}{\mathbf{P}(\gamma,\Gamma)}$. The denominator $\to 1$, so we examine the numerator:

$$\begin{aligned}
\mathbf{P}(t, \gamma, \Gamma) &= \mathbf{P}(t) - \mathbf{P}(t, \neg(\gamma, \Gamma)) \\
&= \mathbf{P}(t) - \mathbf{P}(t, \neg\gamma) - \mathbf{P}(t, \neg\Gamma) + \mathbf{P}(t, \neg\gamma, \neg\Gamma)
\end{aligned}$$

We have: $\mathbf{P}(t, \neg\gamma)/\mathbf{P}(t) \to 0$ by the previous corollary; $\mathbf{P}(t, \neg\Gamma)/\mathbf{P}(t) \to 0$ by induction hypothesis; and $\mathbf{P}(t, \neg\gamma, \neg\Gamma)/\mathbf{P}(t) \leq \mathbf{P}(t, \neg\gamma)/\mathbf{P}(t) \to 0$.

Finally, we can prove Theorem 4. Denoting $\mathbf{P}_0$ the independent probability distribution with the setting of the parameters in the theorem, we have $E_0[R_j] = d_j$ forall j. It suffices to prove that $E_0[R_j \mid \Gamma] = E_0[R_j]$. Consider a ground tuple t of the relation R_j. By linearity of expectation, and since all ground tuples in R have the same marginal probability, we have $E_0[R \mid \Gamma] = N^{arity(R_j)}\mathbf{P}_0[t \mid \Gamma] = N^{arity(R_j)}\mathbf{P}_0[t]\frac{\mathbf{P}_0[t\mid\Gamma]}{\mathbf{P}_0[\Gamma]} \to E_0[R] \times 1$.

D   Proof of Theorem 5

Proof. (of Lemma 1) Consider a subgoal of q of the form $R(x_1, \ldots, x_m, y_1, \ldots, y_p)$. Since the variables $\bar{y}$ are not shared by any other subgoal, this is equivalent to $\exists \bar{y}.R(x_1, \ldots, x_m, \bar{y})$ and, by by Theorem 3, it behaves the same (asymptotically) as an atomic relation symbol $R'(x_1, \ldots, x_m)$ with indepedent tuple probability b_R/N^m. Thus, we will assume w.l.o.g. that there are no variables other than $\bar{x}$ in the body of the query q.

Denote $G_m(x_1, \ldots, x_m)$ the conjunction of all subgoals in q that contain exactly the variables $x_1, \ldots, x_m$ and none others. Thus, $q = G_1(x_1, \ldots, x_k), G_2(x_2, \ldots, x_k), \ldots, G_k(x_k)$. By abuse of notation we will also denote G_m as a set of subgoals, and talk about a relation $R \in G_m$.

In the following we will denote $a_1, \ldots, a_k$ a set of generic constants; recall that $x_1, \ldots, x_k$ are all the variables. For each $m = 0, k$ denote the following:

$$\begin{aligned}
a^{(m)} &= (a_m, a_{m+1}, \ldots, a_k) \\
x_{(m)} &= (x_1, \ldots, x_m) \\
T(x_{(m)}) &= T(x_{(m)}, a^{(m+1)}) \\
G_i(x_{(m)}) &= G_i(x_1, \ldots, x_m, a_{m+1}, \ldots, a_i) \\
G_{(m)}(x_{(m)}) &= G_1(x_{(m)}), \ldots, G_m(x_{(m)})
\end{aligned}$$

For each $m = 0, k$ we define the following reverse inclusion constraint $\delta^{(m)}$:

$$\delta^{(m)} = \forall x_{(m)}.T(x_{(m)}) \Rightarrow G_{(m)}(x_{(m)})$$

Clearly $\delta^k = \delta$. We rewrite $\delta^{(m)}$ as follows. Note that $G_m(x_{(m-1)})$ is a constant subgoal (since G_m does not have any variables in $x_{(m-1)}$), hence we write it as G_m.

$$\delta^{(m)} = \bigwedge_{a_m \in D} (\forall x_{(m-1)} T(x_{(m-1)}) \Rightarrow G_{(m)}(x_{(m-1)}))$$

$$\delta_\perp^{(m)} = \bigwedge_{a_m \in D} (\forall x_{(m-1)} T(x_{(m-1)}) \Rightarrow \perp)$$

$$= \bigwedge_{a_m \in D} (\forall x_{(m-1)} T(x_{(m-1)}) \Rightarrow (G_{(m)}(x_{(m-1)})), G_m)$$

$$= \bigwedge_{a_m \in D} (G_m \wedge \delta^{(m-1)}) \vee \delta_\perp^{(m-1)}$$

We denote $T_{(m-1)} = \exists x_{(m-1)} T(x_{(m-1)})$, and further introduce the notation $B^{(m-1)}$ below, then re-express $\delta^{(m)}$:

$$B^{(m-1)} = (G_m \wedge \delta^{(m-1)}) \vee (\neg G_m \wedge \neg T_{(m-1)})$$

$$\delta^{(m)} = \bigwedge_{a_m \in D} B^{(m-1)}$$

$$\delta^{(0)} = G_1 \vee (\neg G_1 \wedge \neg T)$$

where T stands for $T(\bar{a})$.

Importantly, the events $B^{(m-1)}$ corresponding to different constants $a_m \in D$ are independent, hence the probability $\mathbf{P}_0[\delta^{(m)}]$ we can express it as a product $\prod_{a_m \in D} \mathbf{P}_0[B^{(m-1)}]$.

We start by computing $E_0[T \mid \delta]$, which we write as:

$$E_0[T \mid \delta] = N^k \mathbf{P}_0[T(\bar{a}) \mid \delta] = N^k \frac{\mathbf{P}_0[\delta \mid T(\bar{a})]}{\mathbf{P}_0[\delta]} \mathbf{P}_0[T(\bar{a})]$$

Since $\mathbf{P}_0[T(\bar{a})] = 1 - O(1/N)$ we focus on computing the ratio between the conditional, and unconditional probability of δ Denote $\mathbf{P}_1[-] = \mathbf{P}_0[- \mid T(\bar{a})]$ and observe that $\mathbf{P}_1$ is a tuple-independent distribution that is idential to $\mathbf{P}_0$ on all tuples except the single tuple $T(\bar{a})$. We write $\delta = \delta^{(k)}$, and expand it as $\bigwedge_{a'_k \in D}$ as above, and will separate the conjunct into the case $a'_k = a_k$ and the case $a'_k \neq a_k$: the latter is the same for the a priori $\mathbf{P}_0$ and the conditional $\mathbf{P}_1$.

$$\frac{\mathbf{P}_1[\delta]}{\mathbf{P}_0[\delta]} = \frac{\mathbf{P}_1[B^{(k-1)}] \prod_{a'_k \neq a_k} (\cdots)}{\mathbf{P}_0[B^{(k-1)}] \prod_{a'_k \neq a_k} (\cdots)}$$

$$= \frac{\mathbf{P}_1[G_k] \mathbf{P}_1[\delta^{(k-1)}] + \mathbf{P}_1[\neg G_k] \mathbf{P}_1[\delta_\perp^{(k-1)}]}{\mathbf{P}_0[G_k] \mathbf{P}_0[\delta^{(k-1)}] + \mathbf{P}_0[\neg G_k] \mathbf{P}_0[\delta^{(k-1)}]}$$

$$= \frac{\mathbf{P}_1[G_k] \mathbf{P}_1[\delta^{(k-1)}]}{\mathbf{P}_0[G_k] \mathbf{P}_0[\delta^{(k-1)}] + \mathbf{P}_0[\neg G_k] \mathbf{P}_0[\delta_\perp^{(k-1)}]}$$

Next we prove:

Lemma 6. *(1)* $\delta^0 = \text{\textit{true}}$, $\delta^0_\perp = \neg T(\bar{a})$, *and (2)* $1 \le \mathbf{P}_0[\delta^m]/Pr_0[\delta^m_\perp] \le 1 + O(1/N)$

The second statement follows from the fact that their ratio is $\prod_{a_m} \mathbf{P}_0[\neg \bar{T}(a_m) \vee G(a_m)]/\mathbf{P}_0[\neg T(a_m)]$ and that $\mathbf{P}[\neg T(a_m)]$ domaines asymptotically $\mathbf{P}_0[G(a_m)]$. It follows that $E_0[T \mid \delta] = C(G_k)/d_T$.

Next, we compute $E_0[R \mid \delta]$ Denoting $R(\bar{a})$ a generic tuple we proceed similarly, and note two differences. First $\mathbf{P}_0[R(\bar{a})]$ is no longe 1, but $d_R/N^{arity(R)}$ and we will keep it. Second, $\mathbf{P}_1[-] = \mathbf{P}_0[- \mid R(\bar{a})]$, hence $\mathbf{P}_1[\delta^{(k-1)}_\perp]$ is no longer 0. Instead, here there are two cases. Either there exists a single stratum, i.e. $m = 1$ and then we use the fact that $\mathbf{P}[\delta^0] = 1$ and $\mathbf{P}[\delta^0_\perp] = \mathbf{P}_0[\neg T] = b_T/N^{E(q)}$; in this case we obtain $E_0[R \mid \delta] = C(G_1)/d_T + d_R$. Or there are multiple strata in which case we obtain $E_0[R \mid \delta] = d_R$.