# Understanding Cardinality Estimation using Entropy Maximization

CHRISTOPHER RÉ, University of Wisconsin–Madison
DAN SUCIU, University of Washington, Seattle

Cardinality estimation is the problem of estimating the number of tuples returned by a query; it is a fundamentally important task in data management, used in query optimization, progress estimation, and resource provisioning. We study cardinality estimation in a principled framework: given a set of statistical assertions about the number of tuples returned by a fixed set of queries, predict the number of tuples returned by a new query. We model this problem using the probability space, over possible worlds, that satisfies all provided statistical assertions and maximizes entropy. We call this the Entropy Maximization model for statistics (MaxEnt). In this paper we develop the mathematical techniques needed to use the MaxEnt model for predicting the cardinality of conjunctive queries.

Categories and Subject Descriptors: H.2.4 [**Database Management**]: Systems—*Query processing*

General Terms: Theory

Additional Key Words and Phrases: Cardinality Estimation, Entropy Models, Entropy Maximization, Query Processing

## 1. INTRODUCTION

Cardinality estimation is the process of estimating the number of tuples returned by a query. In relational database query optimization, cardinality estimates are key statistics used by the optimizer to choose an (expected) lowest cost plan. As a result of the importance of the problem, there are many sources of statistical information available to the optimizer, e.g., *query feedback records* [Stillger et al. 2001; Chaudhuri et al. 2008] and *distinct value counts* [Alon et al. 1996], and many models to capture some portion of the available statistical information, e.g., *histograms* [Poosala and Ioannidis 1997; Kaushik and Suciu 2009], *samples* [Haas et al. 1996], and *sketches* [Alon et al. 1999; Rusu and Dobra 2008]; but on any given cardinality estimation task, each method may return a different (and so, conflicting) estimate. Consider the following cardinality estimation task:

> *"Suppose one is given a binary relation $R(A, B)$ along with estimates for the number of distinct values in R.A, R.B, and for the number of tuples in R. Given a query q, how many tuples should one expect to be returned by q?"*

Each of the preceding methods is able to answer the above question with varying degrees of accuracy; nevertheless, the optimizer still needs to make a single estimate, and so, the task of the optimizer is then to choose a single (best) estimate. Although the preceding methods are able to produce an estimate, none is able to say that it is the best estimate (even for our simple motivating example above). In this paper, our goal is to understand the question raised by this observation:

C. Ré is supported by the Air Force Research Laboratory (AFRL) under prime contract no. FA8750-09-C-0181, the National Science Foundation CAREER Award under IIS-1054009, the Office of Naval Research under award no. N000141210041, the University of Wisconsin-Madison, and gifts or research awards from Microsoft, Google, and LogicBlox. Any opinions, findings, and conclusion or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of any of the above sponsors including DARPA, AFRL, ONR or the US government. Suciu is supported by NSF IIS 0915054, NSF IIS 1115188, NSF IIS 1064505, and NSF IIS 0911036.
Author's addresses: C. Ré, Computer Sciences Department, University of Wisconsin–Madison; D. Suciu, Computer Science & Engineering Department, University of Washington, Seattle;
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© YYYY ACM 0362-5915/YYYY/01-ARTA $10.00
DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

*Given some set of statistical information, what is the best cardinality estimate that one can make?* Building on the principle of *entropy maximization*, we are able to answer this question in special cases (including the above example). Our hope is that the techniques that we use to solve these special cases will provide a starting point for a comprehensive theory of cardinality estimation.

Conceptually, our approach to cardinality estimation has two phases: we first build a consistent probabilistic model that incorporates all available statistical information, and then we use this probabilistic model to estimate the cardinality of a query $q$. The standard model used in cardinality estimation is the frequency model [Srivastava et al. 2006]. For example, this model can express that the frequency of the value $a_1$ in $R.A$ is $f_1$, and the frequency of another value $a_2$ in $R.A$ is $f_2$. The frequency model is a probability space over a set of possible tuples. For example, histograms are based on the frequency model. This model, however, cannot express cardinality statistics, such as $\#R.A = 2000$ (the number of distinct values in $A$ is 2000). To capture these, we use a model where the probability space is over the set of possible instances of R, also called *possible worlds*. To make our discussion precise, we consider a language that allows us to make *statistical assertions* which are pairs $(v, d)$ where $v$ is a view (first order query) and $d > 0$ is a real number. An assertion is written $\#v = d$, and its informal meaning is that *"the estimated number of distinct tuples returned by v is d"*. A *statistical program*, $\Sigma = (\bar{v}, \bar{d})$, is a set of statistical assertions, possibly with some constraints. In our language, our motivating question is modeled as a simple statistical program: $\#R = d_R$, $\#R.A = d_A$, and $\#R.B = d_B$. A statistical program defines the statistical information available to the cardinality estimator when it makes its prediction. We give a semantics to this program following prior work [Markl et al. 2005; Srivastava et al. 2006; Kaushik et al. 2009]: our chief desideratum is that our semantic for statistical programs should take into consideration all of the provided statistical information and nothing else. This is the essence of our study: we want to understand what we can conclude from a given set of statistical information without making ad hoc assumptions. Although the preceding desideratum may seem vague and non-technical, as we explain in Section 2, mathematically this can be made precise using the *entropy maximization principle*. In prior work [Kaushik et al. 2009; Ré and Suciu 2010], we showed that this principle allows us to give a semantics to any consistent set of statistical estimates.[1]

Operationally, given a statistical program $\Sigma$, the entropy maximization principle tells us that we are not looking for an arbitrary probability distribution function, but one with a prescribed form. For an arbitrary discrete probability distribution over $M$ possible worlds one needs to specify $M - 1$ numbers; in the case of a binary relation $R(A, B)$ over a domain of size $N$, there are $M = 2^{N^2}$ possible worlds. In contrast, a maximum entropy distribution (MaxEnt) over a program $\Sigma$ containing $t$ statistical assertions is completely specified by a tuple of $t$ parameters, denoted $\bar{\alpha}$. In our motivating question, for example, the maximum entropy distribution is completely determined by three parameters: one for each statistical assertion in $\Sigma$. This raises two immediate technical challenges for cardinality estimation: Given a statistical program $\Sigma$, how do we compute the parameters $\bar{\alpha}$? We call this the *model computation problem*. Then, given the parameters $\bar{\alpha}$ and a query $q$, how does one estimate the number of tuples returned by $q$? We call this the *prediction problem*. In this work, we completely solve this problem for many special cases, including binary relations where $q$ is a full query (i.e., a conjunctive query without projections).

Our first technical result is an explicit, closed-form formula for the expected size of a conjunctive query without projection for a large class of programs called *normal form programs* (NF programs). The formula expresses the expected size of the query in terms of moments of the underlying MaxEnt distribution: the number of moments and their degree depends on the query, and the size of the formula for a query $q$ is $O(|q|)$. As a corollary, we give a formula for computing the expected size of any conjunctive query (with projection) that uses a number of moments that depends on the size of the domain. Next, we show how to extend these results to more statistical programs. For that, we introduce a general technique called *normalization* that transforms arbitrary statistical programs into

[1] Intuitively, a program is consistent if there is at least one probability distribution that satisfies it (see Section 2 for more detail).

normal form programs. A large class of statistical programs are normalized into NF programs, where we can use our estimation techniques. We solve our motivating question with an application of this technique: to make predictions in this model, we normalize it first into an NF program, then express the expected size of any projection-free query in terms of moments of the MaxEnt distribution. By combining these two techniques, we solve size estimation for projection-free queries on a large class of models.

To support prediction, we need to compute both the parameters of the MaxEnt distribution and the moments of the MaxEnt distribution efficiently. The first problem is model computation: given the observed statistics, compute the parameters of the MaxEnt distribution that corresponds to those statistics. This is, in general, a very difficult problem and is intimately related to the problem of learning in *statistical relational models* [Wainwright and Jordan 2008]. We show that for *chain programs* the parameters can be computed exactly, for *hypergraph programs* and *binary relational programs* the parameters can be computed asymptotically (as the domain size $N$ grows to infinity), and for general *relational programs* the parameters can be computed numerically. For the last two methods we have observed empirically that the approximations error is quite low even for relatively small domain sizes (say $N \approx 300$), which makes these approximations useful in practice (especially as input to a numeric solving method). The second problem is: once we have the parameters of the model, compute any given moment. Once the parameters are known, any moment can be computed in time $N^{O(t)}$, where $t$ is the number of parameters of the model, but in some applications this may be too costly. We give explicit closed formulas for approximating the moments, allowing them to be computed in $O(t)$ time.[2] Thus, combining with our previous solution for prediction, we can estimate the expected output size of a projection-free conjunctive query $q$ in time $O(|q|)$.

main tool in deriving asymptotic approximation results is a novel approximation technique, called the *Peaks Approximation*, that approximates the MaxEnt distribution with a convex sum of simpler distributions. In some cases, the Peaks Approximation is very strong: all finite moments of the MaxEnt distribution are closely approximated by the Peaks Approximation. A classical result in probability theory states that, if two finite, discrete distributions agree on all finite moments then they are the same distribution [Shao 2003, pg. 35]. And so, if our approximation were not asymptotic then the Peaks Approximation would not be an approximation – it would be the actual MaxEnt distribution.

#### Outline and Novelty

In Section 2, we discuss the basics of the MaxEnt model. In Section 3, we explain our first technical contribution, *normalization*. In Section 4, we address *prediction* by showing how to estimate the size of a full query in terms of the moments of an MaxEnt model. This section is substantially expanded from our conference version. Then, in Section 5, we discuss the *model computation* problem and solve several special cases using a novel technique, the *Peaks Approximation*, along with full proofs that are new to this version. We discuss an extension of our techniques to histograms in Section 6 (that previously appeared [Kaushik et al. 2009; Ré and Suciu 2010]) including the statement and proof of our result for histograms. We discuss related work in Section 7 and conclude in Section 8.

## 2. THE MAXENT MODEL FOR STATISTICAL PROGRAMS

We introduce basic notations including the queries that we consider. We then review the basic properties of the MaxEnt model. Finally, we give descriptions of the statistical programs that we consider for the remainder of the paper.

*Notation.* We assume a fixed countably infinite set of constants $\mathsf{Const}$. A *schema* is a finite sequence $\bar{R} = \langle R_1, \ldots, R_m \rangle$ where each $R_i$ has a fixed arity $r_i > 0$. An *instance* $I$ (over $\bar{R}$) is a sequence $\langle R_1^I, \ldots, R_m^I \rangle$ such that each $R_i^I$ is a finite relation of arity $r_i$, i.e., a finite subset of $\mathsf{Const}^{r_i}$. We may abuse notation and use $R_i$ to denote both the symbol $R_i$ and its corresponding instance $R_i^I$. We will

[2] We assume here the *unit cost model* [Papadimitriou 1994, pg. 40], i.e., arithmetic operations are constant cost.

often consider a fixed, finite domain $D \subseteq \text{Const}$. In this case, the preceding definitions are modified in a straightforward way, e.g., $\mathrm{R}_i^I \subseteq D^{r_i}$. Throughout this paper, we define $N = |D|$. $N$ will play a central role in our technical development.

A *conjunctive query* over $\bar{R}$ has the form $\exists \bar{y}.\phi(\bar{x}, \bar{y}, \bar{c})$ where $\bar{x}$ and $\bar{y}$ are tuples of variables, $\bar{c}$ is a tuple of constants, and $\phi$ is a conjunction of positive atomic formulas over $\bar{R}$. Here, $\bar{x}$ denotes the *head variables*, whose bindings will be returned by a query when applied to an instance. For example, $q(x) = \exists y_1 \exists y_2.R(x, y_1), S(y_1, y_2, c)$ is a conjunctive query over schema $\langle R, S \rangle$. We assume that $\phi$ is safe in that each variable in $\bar{x}$ is used in some atomic formula. A *full conjunctive query* is a conjunctive query that contains no variables, $\phi(\bar{x})$; a full conjunctive query is, thus, a collection of grounded tuples. We use a standard datalog-style notation to denote such queries [Abiteboul et al. 1995]. A *view* is just another name for a query: we refer to queries and views interchangeably in this paper.

CQ denotes the class of conjunctive queries over a relational schema $\bar{R}$. A *projection query* is a query that contains a single atom without repeated variables. For example, $q(x) :- R(x, y)$ is a projection query, while $q(x) :- R(x, x)$ is not. We also denote projection queries using a named perspective [Abiteboul et al. 1995], e.g., $R_i(A_1, \ldots, A_t)$ then $R_i.A_1A_2$ denotes the projection of $R_i$ onto the attributes $A_1A_2$. To specify statistics for range values, as in a histogram, one needs arithmetic predicates such as $x < y$. To simplify presentation, our queries do not contain arithmetic or inequality predicates. In Section 6, we extend our results to handle arithmetic predicates. Given a view $v$ and a database $I$, we denote by $v(I)$ its output on database $I$. For projection queries in the named perspective, we use the more standard notation $R^I.A_1$ to denote the set of values that appear in attribute $A_1$ of relation $R$ in database $I$.

Let $\Gamma$ be a set of *full inclusion constraints*, i.e., statements of the form $R_i.X \subseteq R_j$, where $X$ is a set of attributes of $R_i$, meaning $\forall \bar{x}.\exists \bar{y}.R_i(\bar{x}, \bar{y}) \Rightarrow R_j(\bar{x})$.

### 2.1. Background: The MaxEnt Model

For a fixed, finite domain $D$ and fixed constraint set $\Gamma$, we denote by $\mathsf{I}(\Gamma)$ the set of all instances over $D$ that satisfy $\Gamma$. The set of all instances over $D$ is $\mathsf{I}(\emptyset)$, which we abbreviate $\mathsf{I}$. A probability distribution on $\mathsf{I}(\Gamma)$ is a tuple of numbers $\bar{p} = (p_I)_{I \in \mathsf{I}(\Gamma)}$ in $[0, 1]$ that sum to 1. We use the notations $p_I$ and $\Pr[I]$ interchangeably in this paper.

A *statistical program* is a triple $\Sigma = (\Gamma, \bar{v}, \bar{d})$, where $\Gamma$ is a set of constraints, $\bar{v} = (v_1, \ldots, v_s)$ and each $v_i$ is a projection query, and $(d_1, \ldots, d_s)$ are positive real numbers. A pair $(v_i, d_i)$ is a *statistical assertion* that we also write as $\#v_i = d_i$. For example, we can assert the cardinality of a relation, $\#R = d_1$ or the cardinality of a single attribute $\#R.A = d_2$.

A probability distribution on $\mathsf{I}(\Gamma)$ *satisfies* a statistical program $\Sigma$ if $\mathbf{E}_{\bar{p}}[|v_i|] = d_i$ for all $i = 1, \ldots, s$. Here $\mathbf{E}_{\bar{p}}[|v_i|]$ denotes the expected value of the size of the view $v_i$, i.e., $\sum_{I \in \mathsf{I}(\Gamma)} |v_i(I)| p_I$. Given a probability distribution $\bar{p}$, we define the estimate that $\bar{p}$ makes for the size of a view (or query) $q$ to be $\mathbf{E}_{\bar{p}}[|q|]$. Our goal is to compute a probability distribution $\bar{p}$ that satisfies the statistical program, then given a query $q$ estimate $\mathbf{E}_{\bar{p}}[|q|]$.

We will let the domain size, $N$, 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 $\lim_{N \to \infty} \mathbf{E}_{\bar{p}^{(N)}}[|v_i|] = d_i$, for $i = 1, \ldots, s$.

Given a program $\Sigma$, we want to determine the most "natural" probability distribution $\bar{p}$ that satisfies $\Sigma$ and we will use it to estimate query cardinalities. In general, there may not exist any probability distribution that satisfies $\Sigma$; in this case, we say that $\Sigma$ is unsatisfiable. We say that a program $\Sigma = (\bar{v}, \bar{d})$ is *satisfiable* if there exists a distribution $\bar{p}$ such that for $i = 1, \ldots, s$, $\mathbf{E}_{\bar{p}}[|v_i|] = d_i$ and *unsatisfiable* otherwise.[3] On the other hand, there may exist many solutions. To choose a canonical one, we apply the principle of Maximum Entropy (MaxEnt).

[3]Using a compactness argument, we show in Appendix A.1 that if a program is satisfiable, there is at least one distribution that maximizes entropy. In general, we are unaware of a simple procedure to check satisfiability of a program. For the programs that we consider in this paper, these checks are straightforward.

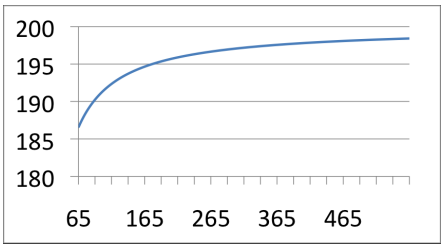


Fig. 1. A graph that plots the domain size (on $x$-axis) versus $\mathbf{E}[|R.AC|]$ ($y$-axis) for the program $R(A, B, C)$: $\#R = 200$, $\#R.A = 20$, $\#R.B = 30$, $\#R.C = 40$.

*Definition* 2.1. A probability distribution $\bar{p} = (p_I)_{I \in \mathsf{I}(\Gamma)}$ is a MaxEnt distribution associated to $\Sigma$ if the following two conditions hold: (1) $\bar{p}$ satisfies $\Sigma$, and (2) it has the maximum entropy among all distributions that satisfy $\Sigma$, where the entropy of $\bar{p}$ is $H(\bar{p})$ is defined as the quantity $-\sum_{I \in \mathsf{I}(\Gamma)} p_I \log p_I$.

We refer to a MaxEnt distribution as *the* MaxEnt model, since, as we later show, it is uniquely defined. Let $\bar{p}(\Sigma)$ be the distribution associated to $\Sigma$, then we define $\Pr_\Sigma$ to be $\Pr_{\bar{p}(\Sigma)}$ and $\mathbf{E}_\Sigma$ to be $\mathbf{E}_{\bar{p}(\Sigma)}$.

For a simple illustration, consider the following program on the relation $R(A, B, C)$: $\#R = 200$, $\#R.A = 20$, $\#R.B = 30$, $\#R.C = 40$. 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 $\#R.AC$, i.e., the number of distinct values of pairs $AC$. Clearly this number can be anywhere between 40 and 200, but currently there does not exist a principled approach for query optimizers to estimate the number of distinct pairs $AC$ from the other four statistics. The MaxEnt 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 $\mathbf{E}_{\bar{p}}[|R.AC|] = \sum_{I \subseteq D^3} p_I |R^I.AC|$. This estimate will certainly be between 30 and 200; it will depend on $N$, which is an undesirable property. Ideally, one could compute this estimate for each fixed $N$ in a closed form. We are unable to do this (except by using directly the definition of the expected value, $\mathbf{E}_{\bar{p}}[|v|] = \sum_{I \in \mathsf{I}(\Gamma)} |v(I)| p_I$, whose size is exponential in $N$), but we are able to solve these equations numerically. To make the computations more tractable while removing the sensitivity on $N$, a sensible thing to do is to let $N$ grow to infinity, and compute the limit of $\mathbf{E}_{\bar{p}}[|R.AC|]$. In Figure 1, we plot $\mathbf{E}_{\bar{p}}[|R.AC|]$ as a function of the domain size ($N$). Interestingly, it very quickly goes to 200, even for small values of $N$. Thus, the MaxEnt model offers a principled and uniform approach to query size estimation.

To describe the general form of a MaxEnt distribution, we need some definitions. Fix a program $\Sigma = (\Gamma, \bar{v}, \bar{d})$, and so a set of constraints $\Gamma$ and views $\bar{v} = (v_1, \ldots, v_s)$.

*Definition* 2.2. The *partition function* for $\Sigma = (\Gamma, \bar{v}, \bar{d})$ is the following polynomial $T^\Sigma$ with $s$ variables $\bar{x} = (x_1, \ldots, x_s)$:

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

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

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

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

We write $T$ instead of $T^{\Sigma}$ when $\Gamma, \bar{v}$ are clear from the context (notice that $T$ does not depend on $\bar{d}$), and express it 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 $\Gamma$ and for which $|v_i(I)| = k_i$, for all $i = 1, \ldots, s$. Our technical developments need a function defined by the compact form of the partition function that we call a *term function* that is denoted $t^{\Sigma}$ and is defined by:

$$t^{\Sigma}(\bar{x}; \bar{k}) = C_\Gamma(N, k_1, \ldots, k_s) x_1^{k_1} \cdots x_s^{k_s} \text{ so that } T^{\Sigma}(\bar{x}) = \sum_{\bar{k}} t^{\Sigma}(\bar{x}; \bar{k})$$

The following is a key characterization of MaxEnt distributions.

Theorem 2.3. *[Jaynes 2003, page 355] Let $\Sigma$ be a statistical program. For any probability distribution $\bar{p}$ that satisfies the statistics $\Sigma$ the following holds: $\bar{p}$ is a* MaxEnt *distribution iff there exists a tuple of parameters $\bar{\alpha}$ such that $\bar{p}$ is given by the Equation (1) (equivalently: $\bar{p}$ is associated to $(\Sigma, \bar{\alpha})$).*

We refer to Jaynes [Jaynes 2003, page 355] for a full proof; the "only if" part of the proof is both simple and enlightening, and so we reproduce it here:

Proof. The "only if" direction is very simple to derive by using Lagrange multipliers to solve:

$$F_0 = \sum_{I \in \mathsf{I}} p_I - 1 = 0 \tag{2}$$

$$\forall i = 1, \ldots, s: \quad F_i = \sum_{I \in \mathsf{I}} |v_i(I)| p_I - d_i = 0 \tag{3}$$

$$H = \text{maximum, where} \quad H = \sum_{I \in \mathsf{I}} -p_I \log p_I \tag{4}$$

According to the method, one has to introduce $s + 1$ additional unknowns, $\lambda_0, \lambda_1, \ldots, \lambda_s$: a MaxEnt distribution is a solution to a system of $|\mathsf{I}| + s + 1$ equations consisting of Eq.(2), (3), and the following $|\mathsf{I}|$ equations:

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

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

Note that in Theorem 2.3 the parameters $\bar{\alpha}$ are not necessarily unique. However, they are always unique if the views $\bar{v}$ are *affinely independent*. Call the $m$ views $\bar{v}$ *affinely dependent* over a set of instances $\mathsf{I}(\Gamma)$ if there exist $m + 1$ real numbers $\bar{c}, d$, not all zero, such that:

$$\forall I \in \mathsf{I}(\Gamma). \quad \sum_{j=1,\ldots,s} |v_j(I)| c_j = d$$

We say $\bar{v}$ is affinely independent over $\mathsf{I}(\Gamma)$ if no such $\bar{c}, d$ exist.

Theorem 2.4. *Let $\Sigma = (\Gamma, \bar{v}, \bar{d})$ be a satisfiable statistical program where $\bar{v}$ is affinely independent over $\mathsf{I}(\Gamma)$, then there is a unique tuple of parameters $\bar{\alpha}$ that satisfies $\Sigma$ and maximizes entropy.*

Minor variants of this theorem have previously appeared in the literature [Wainwright and Jordan 2008, §3.2]. Nevertheless, we include a proof for completeness in Section A. As a trivial example of an affinely dependent statistical program, consider two views $v_1$ and $v_2$ that have identical

definitions: then for every instance $|v_1(I)| = |v_2(I)|$, and, in this case, the MaxEnt model is not uniquely defined: the parameters $\alpha_1, \alpha_2$ can be varied while keeping their product $\alpha_1\alpha_2$ constant. As another example, one may have a statistical program $\Sigma$ consisting of a histogram with $m$ buckets, defined by the views $v_1, \ldots, v_m$, and the total number of elements in the histograms, $v_0$: then $|v_0(I)| = |v_1(I)| + \ldots + |v_m(I)|$. In all the statistical programs that we consider in this paper, one can always choose a subset of the views that are affinely independent, hence, from now on we will assume without loss of generality that, for any program $\Sigma = (\Gamma, \bar{v}, \bar{d})$ we consider, $\bar{v}$ is affinely independent over $\mathsf{I}(\Gamma)$.[4] Therefore, the parameters that maximize the entropy are uniquely defined, justifying the term "*the* MaxEnt Model".

We illustrate with an example:

*Example* 2.5. **The Binomial Model** Consider a relation $R(A, B)$ and the statistical assertion $\#R = d$. The partition function is the binomial, $T(x) = \sum_{k=0,\ldots,N^2} \binom{N^2}{k} x^k = (1+x)^{N^2}$. We claim that the MaxEnt model is the following probability distribution: randomly insert each tuple in $R$ independently, with probability $p = d/N^2$. This is the Binomial Model, given by $\Pr[I] = p^k(1-p)^{N^2-k}$. To check that this is a MaxEnt distribution, rewrite it as $\Pr[I] = \omega\alpha^k$. Here $\alpha = p/(1-p)$ is the *odds* of a tuple, and $\omega = (1-p)^{N^2} = \Pr[I = \emptyset]$. This is indeed a MaxEnt distribution by Theorem 2.3. Asymptotic query evaluation on a generalization of this distribution to multiple tables has been studied [Dalvi et al. 2005]. □

In this example, $\alpha$ is the odds of a particular tuple. In general, the MaxEnt parameters may not have a simple probabilistic interpretation. They do, however, determine all the moments of the distribution as we explain in the following proposition[5]:

Proposition 2.6. *Let $\Sigma$ be a partition function for $\Sigma = (\Gamma, \bar{v}, \bar{d})$ with parameters $\alpha_v$ then for $v \in \bar{v}$ we have:*

$$T^{\Sigma}(\bar{\alpha}) \times \mathbf{E}[|v|^k] = \left(\alpha_v \frac{\partial}{\partial \alpha_v}\right)^k T^{\Sigma}(\bar{\alpha})$$

*where $(\alpha_v \frac{\partial}{\partial \alpha_v})^k$ denotes that the operator $\theta = \alpha_v \frac{\partial}{\partial \alpha_v}$ is applied k times, and*

$$T^{\Sigma}(\bar{\alpha}) \times \mathbf{E}[\langle |v| \rangle_{(k)}] = \alpha_v^k \frac{\partial^k}{\partial^k \alpha_v} T^{\Sigma}(\bar{\alpha})$$

Here $\langle x \rangle_{(k)} = x(x-1)\cdots(x-k+1)$ denotes the falling factorial. The proof is straightforward: apply the operators directly to the partition function, $T^{\Sigma}$ in compact form and use linearity. Since MaxEnt distributions are polynomials, computing derivatives is straightforward – but possibly expensive.

*Example* 2.7. Consider the binomial model: $T(x) = \sum_{k=0}^{N} \binom{N}{k} x^k = (1+x)^N$. This model can be viewed as a distribution over the size of a randomly chosen subset $v \subseteq [N]$. Consider the expected size of $v$, $\mathbf{E}[|v|]$. By definition:

$$\mathbf{E}[|v|] = \frac{\sum_{k=0}^{N} \binom{N}{k} k x^k}{T(x)}$$

The proposition above gives us a convenient way to compute $\mathbf{E}[|v|]$ as follows: $T(x)\mathbf{E}[|v|] = (x\frac{\partial}{\partial x})(1+x)^N = Nx(1+x)^{N-1}$. It follows that $\mathbf{E}[|v|] = Nx/(1+x)$. In similar way, the value of $\mathbf{E}[|v|^2]$ can be found by applying $\left(x\frac{\partial}{\partial x}\right)$ twice.

[4] We prove that all programs that we define below are affinely independent in Appendix A.2.

[5] A *moment* is a real number that measures some random variable of a probability distribution. The $k^{\text{th}}$ (raw) moment of a random variable is $X$ is $\mathbf{E}[X^k]$. The $k^{\text{th}}$ factorial moment of $X$ is $\mathbf{E}[\langle X \rangle_{(k)}]$. A discrete distribution is uniquely determined by either its factorial moments or its raw moments [Shao 2003, pg. 35]. We also consider moments of more than one random variable, e.g., $\mathbf{E}[X\langle Y \rangle_{(2)}]$ denotes the joint moment of $X$ and the 2nd factorial moment of $Y$.

### 2.2. A Normal Form for Statistical Programs

We define a normal form for statistical program. We say an assertion is on a base table if it is of the form $\#R = c$ for some relational symbol $R$ and constant $c$.

*Definition* 2.8. $\Sigma$ is in *normal form* (NF) if all statistical assertions are on base tables; otherwise, it is in *non-normal form* (NNF).

For illustration, consider the relation $R(A_1, A_2)$. The program $\#R = 20, \#R.A_1 = 10$, and $\#R.A_2 = 5$ where $\Gamma = \emptyset$ is in NNF. Consider three relation names $S(A_1, A_2)$, $R_1(A_1)$, $R_2(A_2)$. The program with constraints $S.A_i \subseteq R_i$ for $i = 1, 2$ and statistical assertions $\#S = 20, \#R_1 = 10, \#R_2 = 5$ is in NF.

Any NF program without constraints consists of independent binomial models, one for each base table. When there are constraints, then an NF program may not be a binomial. Still, we have found NF programs to be easier to reason about than a NNF program. We will prove in Section 3 that every program can be translated into a statistical program in normal form.

### 2.3. Important Programs

We describe two classes of programs that are central to this paper: *relational programs* and *hypergraph programs*.

#### 2.3.1. Relational Statistical Programs

*Definition* 2.9. Fix a single relation name $R(A_1, \ldots, A_m)$. A *relational program* is a program $\Sigma = (\Gamma, \bar{v}, \bar{d})$ where $\Gamma = \emptyset$ and every statistical assertion is of the form $\#R.X = d$ for $X \subseteq \{A_1, \ldots, A_m\}$.

That is, all views are projections are queries over one table, and there are no constraints. Relational programs are in NNF.

A relational program is called *simple* it consists of $m + 1$ assertions: $\#R.A_i = d_i$ for $i = 1, \ldots, m$, and[6] $\#R = d_R$. We always order the parameters and assume without loss of generality that $d_1 \leq d_2 \leq \ldots d_m \leq d_R$. Our motivating example in the introduction is a simple relational program of arity 2.

We now give the partition function for a simple relational program. For a fixed $N \geq 0$, let $r(\bar{k}, l) = r(k_1, \ldots, k_m, l)$ be the number of relational instances $I$ with schema $R(A_1, \ldots, A_m)$ where the attribute of each domain is of size $N$ such that $|R^I| = l$ and $|R^I.A_i| = k_i$ for $i = 1, \ldots, k$.

Proposition 2.10. *Consider a simple relational program $\Sigma_{Rm}$ of arity m For $i = 1, \ldots, m$, let $\alpha_i$ be the parameter associated with the assertion $\#R.A_i = d_i$. Let $\gamma$ be the parameter associated with the assertion $|R| = d_R$. The partition function $T^{\Sigma_{Rm}}$ associated to $\Sigma_{Rm}$ is:*

$$T^{\Sigma_{Rm}}(\bar{\alpha}, \gamma) = \sum_{\bar{k},l} r(\bar{k}, l)\gamma^l \prod_{i=1}^{m} \binom{N}{k_i} \alpha_i^{k_i}$$

The function $r(\bar{k}, l)$ is difficult to compute. One can show, using the inclusion/exclusion principle, that for $m = 2$:

$$r(k_1, k_2, l) = \sum_{\substack{j_1 = 0, \ldots, k_1 \\ j_2 = 0, \ldots, k_2}} (-1)^{j_1 + j_2} \binom{k_1}{j_1} \binom{k_2}{j_2} \binom{(k_1 - j_1)(k_2 - j_2)}{l}$$

This generalizes to arbitrary $m$. To the best of our knowledge, there is no simple closed form for $r$: we circumvent computing $r$ using normalization, in Section 3.

#### 2.3.2. Hypergraph Statistical Programs

[6] $\#R$ is equivalent to $\#R.A_1A_2 \ldots A_m$.

*Definition* 2.11. Fix a set of relation names $R_1, R_2, \ldots, R_m$. A hypergraph program consists of $\Sigma = (\Gamma, \bar{v}, \bar{d})$, where $\Sigma$ has one statistical assertion $\#R_i = d_i$ for every relation name $R_i$, and $\Gamma$ consists of inclusion constraints of the form $R_i.X \subseteq R_j$, where $X$ is a subset of the attributes of $R_i$.

A hypergraph program is in NF. If there are no constraints, then a hypergraph program consists of $m$ independent binomial models. The addition of constraints changes the model considerably.

We consider two important special cases of hypergraph programs in this paper: *chain programs* and *simple hypergraph programs*.

*Chain Programs.* The first is a chain program. Fix $m$ relation names: $R_1(A_1, \ldots, A_m)$, $R_2(A_2, \ldots, A_m)$, …, $R_m(A_m)$. A *chain program* of size $m$, $\Sigma_{Cm}$, is a hypergraph program where $\Gamma$ the set of constraints is defined as

$$\Gamma = \{R_{i-1}.A_iA_{i+1}\ldots A_m \subseteq R_i \mid i = 2, \ldots, m\}$$

For example, $\Sigma_{C2}$ is the following program on $R_1(A_1, A_2)$ and $R_2(A_2)$: $\#R_1 = d_1$, $\#R_2 = d_2$, and $R_1.A_2 \subseteq R_2$.

Proposition 2.12. *(Chain Partition Function) Let $\Sigma_{Cm}$ be a chain program of size $m \geq 1$. Denote the parameters of $\Sigma_{Cm}$ as $\bar{\alpha} = (\alpha_1, \ldots, \alpha_m)$. Then its partition function satisfies the recursion:*

$$\begin{aligned} T^{\Sigma_{C1}}(\alpha_1) &= (1+\alpha_1)^N \\ T^{\Sigma_{Cj+1}}(\alpha_1, \ldots, \alpha_{j+1}) &= \left(1 + \alpha_{j+1} T^{\Sigma_{Cj}}(\alpha_1, \ldots, \alpha_j)\right)^N \end{aligned}$$

*for $j = 1, 2, \ldots, m-1$.*

To prove this proposition, apply the binomial theorem inductively. The partition function $T^{\Sigma_{Cm}}$ is sometimes referred to as a *cascading binomial* [Dalvi et al. 2005].

*Example* 2.13. Consider the schema $R_1(A_1, A_2), R_2(A_2)$. The chain program $\Sigma_{C2}$ is $\#R_1 = d_1$, $\#R_2 = d_2$, and $R_1.A_2 \subseteq R_2$, and its partition function is:

$$T^{\Sigma_{C2}}(\bar{\alpha}) = (1 + \alpha_2(1+\alpha_1)^N)^N$$

Given $\bar{d} = (d_1, d_2)$, we need to find the parameters $\alpha_1, \alpha_2$ for which the probability distribution defined by $T^{\Sigma_{C2}}$ has $E[|R_1|] = d_1$ and $E[|R_2|] = d_2$.

We now show that the solutions are $\alpha_1 = \frac{d_1}{d_2 N - d_1}$ and $\alpha_2 = \frac{d_2}{N - d_2}(1+\alpha_1)^{-N}$. We verify that:

$$\begin{aligned} \mathbf{E}[|R_2|] &= \frac{1}{T^{\Sigma_{c2}}}\, \alpha_2 \frac{\partial}{\partial \alpha_2}\, T^{\Sigma_{C2}} = N \frac{\alpha_2(1+\alpha_1)^N}{1 + \alpha_2(1+\alpha_1)^N} = N \frac{d_2}{(N-d_2)+d_2} = d_2 \\ \mathbf{E}[|R_1|] &= \frac{1}{T^{\Sigma_{c2}}}\, \alpha_1 \frac{\partial}{\partial \alpha_1}\, T^{\Sigma_{C2}} = N \frac{\alpha_2(1+\alpha_1)^N}{1 + \alpha_2(1+\alpha_1)^N} N \frac{\alpha_1}{1+\alpha_1} = \mathbf{E}[|R_2|] N \frac{\alpha_1}{1+\alpha_1} = d_2 N \frac{d_1}{N d_2} = d_1 \end{aligned}$$

These results hold for any domain $N$ that is large enough (here $N > d_2$ and $d_2 N > d_1$). Observe that no limits are needed.

*Simple Hypergraph Programs.* The second special case is the following. A *simple hypergraph program* of size $m$ is a hypergraph program over $S(A_1, \ldots, A_m), R_1(A_1), \ldots, R_m(A_m)$, where the constraints are $S.A_i \subseteq R_i$ for $i = 1, \ldots, m$. We denote by $\Sigma_{Hm}$ a simple hypergraph program of size $m$, and will refer to it, with some abuse, as a hypergraph program. Its partition function is:

Proposition 2.14 (Hypergraph Partition Function). *Given a hypergraph program $\Sigma_{Hm}$, let $\bar{\alpha}$ be a tuple of $m$ parameters (one for each $R_i$) and $\gamma$ be the parameter associated with the assertion on $S$. Then, the partition function is given by:*

$$T^{\Sigma_{Hm}}(\bar{\alpha}, \gamma) = \sum_{\bar{k}} t^{\Sigma_{Hm}}(\bar{\alpha}, \gamma; \bar{k}) \text{ where } t^{\Sigma_{Hm}}(\bar{\alpha}, \gamma; \bar{k}) = (1+\gamma)^{\prod_{i=1}^m k_i} \times \prod_{i=1}^m \binom{N}{k_i} \alpha_i^{k_i}$$

This partition function corresponds to a simple two-stage random process: select random sets $R_i$ from the domain using a binomial distribution, then select a random subset $S$ of edges (hyperedges) from their cross product $R_1 \times \cdots \times R_m$ using another binomial distribution. Note that the term function is simpler than in Proposition 2.10. This term function will play a central role in our later technical developments.

*Example* 2.15. The hypergraph program $\Sigma_{H2}$ is over three relations, $S(A_1, A_2)$, $R_1(A_1)$, and $R(A_2)$, two constraints $S.A_1 \subseteq R_1$, $S.A_2 \subseteq R_2$, and three statistical assertions: $\#R_1 = d_1$, $\#R_2 = d_2$, $\#S = d_S$. Denoting $\alpha_1$, $\alpha_2$, and $\gamma$ the parameters of the MaxEnt model, we have:

$$T^{\Sigma_{H2}}(\alpha_1, \alpha_2, \gamma) = \sum_{k_1,k_2} \binom{N}{k_1}\binom{N}{k_2} \alpha_1^{k_1} \alpha_2^{k_2} (1+\gamma)^{k_1 k_2}$$

This expression is much simpler than that in Prop. 2.10, but it still does not have a closed form. To compute moments of this distribution (needed for expected values) one needs sums of $N^2$ terms. The difficulty comes from $(1+\gamma)^{k_1 k_2}$: when $k_1 k_2 \gamma = o(1)$, this term is $O(1)$ and the partition function behaves like a product of two binomials, but when $k_1 k_2 \gamma = \Omega(1)$ it behaves differently.

### 2.4. Problem Definitions

We study two problems in this paper. One is the *model computation problem*: given a statistical program $\Sigma = (\Gamma, \bar{v}, \bar{d})$, find the parameters $\bar{\alpha}$ for the MaxEnt model such that $\bar{\alpha}$ satisfies $\Sigma$. The other is the *prediction* problem, given the parameters of a model and a query $q(\bar{x})$, compute $\mathbf{E}[|q(\bar{x})|]$ in the MaxEnt distribution. We first discuss a technique, normalization, that is useful to attack both problems.

## 3. NORMALIZATION

We give here a general procedure for converting any NNF statistical program $\Sigma$ into an NF program with additional inclusion constraints. In fact, this theorem is the reason why we consider inclusion constraints as part of our statistical programs.

Theorem 3.1 below shows one step of the normalization process: how to replace a statistical assertion on a projection with a statistical assertion on a base table plus one additional inclusion constraint. Repeating this process normalizes $\Sigma$.

We describe the notation in the theorem. Let $\bar{R} = (R_1, \ldots, R_m)$. Let $\bar{v}$ be a set of $s$ projection views, and assume that $v_s$ is not a base relation. Thus, the statistic $\#v_s = d_s$ ensures that the program is in NNF. Let $Q$ be a new relational symbol of the same arity as $v_s$, and set $\bar{R}' = \bar{R} \cup \{Q\}$, $\Gamma' = \Gamma \cup \{v_s \subseteq Q\}$. Replace the statistical assertion $\#v_s = d_s$ with $\#Q = d'_s$ (where the number $d'_s$ is computed as described below). Denote $a = \text{arity}(Q)$. Denote $\bar{w}$ the set of views obtained from $\bar{v}$ by replacing $v_s$ with $Q$.

Let us examine the MaxEnt distributions for $(\Gamma, \bar{v})$ and for $(\Gamma', \bar{w})$. Both have the same number of parameters ($s$). The outcome of the former is a database instance with $m$ relations: $R_1, \ldots, R_m$; the latter has as an outcome a database instance with $m+1$ relations: $R_1, \ldots, R_m, Q$. Consider a MaxEnt distribution for $(\Gamma', \bar{w})$, and examine what happens if we compute the marginal distribution over $R_1, \ldots, R_m$. As we show below, the resulting distribution is another MaxEnt distribution[7], for $(\Gamma, \bar{v})$. More precisely,

Theorem 3.1 (Normalization). *Consider a* MaxEnt *distribution for* $(\Gamma', \bar{w})$*, with parameters* $\beta_1, \ldots, \beta_s$ *and outcomes* $R_1, \ldots, R_m, Q$*. Then the marginal distribution over* $R_1, \ldots, R_m$ *is a* MaxEnt *distribution for* $(\Gamma, \bar{v})$*, with parameters given by* $\alpha_i = \beta_i$ *for* $i = 1, \ldots, s-1$*, and* $\alpha_s = \frac{\beta_s}{1+\beta_s}$*. In*

[7]Given a two outcomes $x, y$ with domain $\mathbb{D}$ and probability distribution given by Pr, the marginal distribution of $x$ is the probability function defined by $\Pr_x[x = d] = \sum_{d' \in \mathbb{D}} \Pr[x = d, y = d']$ where $d \in \mathbb{D}$.

*addition, the following relations hold between the partition functions T for* $(\Gamma, \bar{v})$ *and U for* $(\Gamma', \bar{w})$*:*

$$T(\bar{\alpha}) = \frac{U(\bar{\beta})}{(1+\beta_s)^{N^a}} \tag{5}$$

*Finally, the following relationship holds between the expected sizes of the views in the statistical programs:*

$$\begin{aligned} \mathbf{E}_T[|v_s|] &= N^a\alpha_s + (1-\alpha_s)\mathbf{E}_U[|Q|] \\ \mathbf{E}_T[|v_i|] &= \mathbf{E}_U[|w_i|] \quad \textit{for } i = 1, \ldots, s-1 \end{aligned} \tag{6}$$

Proof. We show Equation 5. Equation 6 follows immediately, by direct derivation. Denote $\mathsf{I} = \mathsf{I}(\Gamma)$ the set of $\bar{R}$-instances that satisfy the constraint $\Gamma$, and $\mathsf{J} = \mathsf{J}(\Gamma')$ the set of $\bar{R}'$ instances that satisfy $\Gamma'$. An instance $J \in \mathsf{J}$ has $m+1$ relations: we write $I = (I_1, \ldots, I_m)$ for the first $m$ and denote $K$ the $m+1$'st relation, thus $J = (I, K)$ (and one can check that $I \in \mathsf{I}$, because the constraints $\Gamma'$ extend $\Gamma$). The additional constraint in $\Gamma'$ is $v_s(I) \subseteq K$. For every instance $J = (I, K)$, the relation $K$ uniquely splits into $K = L \cup P$, where $L = v_s(I)$ and $P = K - v_s(I)$. Note that the sole constraint on $P$ is $P \subseteq D^a - v_s(I)$. Moreover, $|w_s(J)| = |K| = |v_s(I)| + |P|$, and $|w_i(J)| = |v_i(I)|$ for $i = 1, \ldots, s-1$, thus:

$$\begin{aligned} U(\bar{\beta}) &= \sum_{J\in\mathsf{J}} \prod_{i=1}^{s} \beta_i^{|w_i(J)|} = \sum_{I\in\mathsf{I}} \sum_{P\subseteq D^a - v_s(I)} \beta_s^{|v_s(I)|+|P|} \cdot \prod_{i=1}^{s-1} \beta_i^{|v_i(I)|} \\ &= \sum_{I\in\mathsf{I}} \left( \prod_{i=1}^{s-1} \beta_i^{|v_i(I)|} \right) \cdot \beta_s^{|v_s(I)|} \cdot (1+\beta_s)^{N^a - |v_s(I)|} \\ &= \sum_{I\in\mathsf{I}} \left( \prod_{i=1}^{s-1} \beta_i^{|v_i(I)|} \right) \cdot \frac{\beta_s^{|v_s(I)|}}{(1+\beta_s)^{|v_s(I)|}} \cdot (1+\beta_s)^{N^a} \\ &= \sum_{I\in\mathsf{I}} \prod_{i=1}^{s} \alpha_i^{|v_i(I)|} \cdot (1+\beta_s)^{N^a} = T(\bar{\alpha}) \cdot (1+\beta_s)^{N^a} \end{aligned}$$

□

We give some examples of how the normalization theorem is used. Intuitively, the last equation tells us how to set the expected sized $d'_s$ of $Q$ to obtain the same distributions, namely $d'_s = (d_s - N^a\alpha_s)/(1-\alpha_s)$.

*Example* 3.2. **The** $A, R$**-Model (Cascading Binomials)** Consider two statistical assertions on $R(A, B)$: $\#R = d_1$ and $\#R.A = d_2$. Our goal is to compute the parameters $\alpha_1, \alpha_2$ of the MaxEnt model from the two statistics $d_1, d_2$. This program is not normalized: we use Theorem 3.1 to normalize it, compute its parameters $\beta_1, \beta_2$, then obtain $\alpha_1, \alpha_2$. To normalize, add a new relation symbol $Q(A)$, the constraint $R.A \subseteq Q$, and make the following two statistical assertions, $\#Q = c$, $\#R = d_1$; the new constant $c$ is an unknown (it is given by Eq.(6)). This is a chain program, and we gave its solution in Example 2.13: $\beta_1 = d_1/(cN - d_1)$ and $\beta_2 = c/(N-c)(1+\beta_1)^{-N}$. By Theorem 3.1, the parameters of the non-normalized model are $\alpha_2 = \beta_2/(1+\beta_2)$, $\alpha_1 = \beta_1$, and the new statistics, $c$, is given by Eq.(6): $c = N\alpha_2 + (1-\alpha_2)d_2$. Thus, we have a non-linear system of equations with three unknowns,

$\alpha_1, \alpha_2, c$. To solve it, we first derive the following approximation: $\alpha_2 \approx \frac{c}{N} e^{-d_1/c}$. Indeed:

$$\begin{aligned}\alpha_2 = \frac{\beta_2}{1+\beta_2} &= \frac{c}{(N-c)(1+\beta_1)^N}\left(1 + \frac{c}{(N-c)(1+\beta_1)^N}\right)^{-1} \\ &= \frac{c}{(N-c)(1+\beta_1)^N}\frac{(N-c)(1+\beta_1)^N}{(N-c)(1+\beta_1)^N + c} \\ &= \frac{c}{(N-c)(1+\beta_1)^N + c}\end{aligned}$$

Noting that $\lim_{N\to\infty}(1+\beta_1)^N = e^{d_1/c}$, we obtain $\lim_{N\to\infty} N\alpha_2 = \lim_{N\to\infty} \frac{Nc}{(N-c)(1+\beta_1)^N + c} = ce^{-d_1/c}$. Thus, assuming $N$ large enough, we will approximate $\alpha_2 \approx \frac{c}{N} e^{-d_1/c}$. This implies $c = N\alpha_2 + (1-\alpha_2)d_2 \approx ce^{-d_1/c} + d_2$. This is a transcendental equation in $c$, which first we rewrite as $e^{-d_1 v} = 1 - d_2 v$, where $v = 1/c$, then further substitute $t = d_1 v + \frac{d_1}{d_2}$, so that $v = d_1^{-1}(t - \frac{d_1}{d_2})$, obtaining the following:

$$e^{-(t-\frac{d_1}{d_2})} = \frac{d_2}{d_1} t \implies t = \mathsf{W}\left(\frac{d_1}{d_2} e^{-\frac{d_1}{d_2}}\right) \implies v = \frac{\mathsf{W}\left(\frac{d_1}{d_2} e^{-\frac{d_1}{d_2}}\right)}{d_1} - \frac{1}{d_2}$$

where $\mathsf{W}$ is the Lambert-$\mathsf{W}$ (multi-)function defined as $\mathsf{W}(x)e^{\mathsf{W}(x)} = x$ [Corless et al. 1997]. $\mathsf{W}$ is a function (not a multifunction) for positive reals, and $W(xe^{-x}) = x$ occurs only at $x = 0$, thus $v > 0$ for all $d_1, d_2 > 0$. This gives us an explicit solution $c = 1/v$ for $c$ in terms only of the two statistics $d_1, d_2$, and, furthermore, we obtain the parameters $\alpha_2 \approx \frac{c}{N} e^{-d_1/c}$ and $\alpha_1 = d_1/(cN - d_1)$.

*Example* 3.3. To appreciate the power of normalization, we will illustrate on the NNF program on $R(A, B)$: $\#R.A = d_1$, $\#R.B = d_1$, and $\#R = d$. Let $\alpha_1, \alpha_2, \gamma$ be the associated parameters of MaxEnt. Its partition function $T(\alpha_1, \alpha_2, \gamma)$ is a complicated expression given by Prop.2.10. The NF Program has three relations $R_1(A_1)$, $R_2(A_2)$ and $R(A_1, A_2)$, statistics $\#R_1 = c_1, \#R_2 = c_2, \#R = c$, and constraints $R.A_1 \subseteq R_1$, $R.A_2 \subseteq R_2$. Its partition is $U(\beta_1, \beta_2, \gamma) = \sum_{k_1,k_2} \binom{N}{k_1}\binom{N}{k_2}\beta_1^{k_1}\beta_2^{k_2}(1+\gamma)^{k_1 k_2}$ (see Example 2.15). After applying the normalization theorem twice, we obtain the following identity:

$$T(\alpha_1, \alpha_2, \gamma) \;=\; (1+\beta_1)^{-N}(1+\beta_2)^{-N} U(\beta_1, \beta_2, \gamma)$$

where $\alpha_i = \beta_i/(1+\beta_i)$ for $i = 1, 2$. Moreover, $d_i = N\alpha_i + (1-\alpha_i)c_i$ for $i = 1, 2$ and $d = c$. This translation allows us to do predictions for the NNF program by reduction to the (more manageable) NF hypergraph program. This justifies the normalization theorem, and our interest in hypergraph programs.

After normalization, the challenge that remains is to compute the values $\bar{c}$ and the $\bar{\beta}$ parameters that yield $\bar{c}$ according to the normalized model.

## 4. PREDICTION

In this section, we describe how to estimate the size of a projection-free conjunctive query $q$ on a hypergraph program. Then using normalization, we show how to estimate the expected size of a query on a relational program. Throughout this section we assume that the parameters of the model are given: we discuss in the next section how to compute these parameters from given a statistical program.

Recall from Section 2 that a conjunctive query $q$ has the form $\exists \bar{y}.\phi(\bar{x}, \bar{y}, \bar{c})$. We write $q(\bar{x})$ to highlight the head variables $\bar{x}$. Our technique is to rewrite $\mathbf{E}[|q(\bar{x})|]$ in terms of the moments of the MaxEnt distribution. We first reduce computing $\mathbf{E}[|q(\bar{x})|]$ to computing $\Pr[q']$ for several Boolean queries $q'$ (i.e. queries without head variables). Then, we provide an explicit, exact formula for $\Pr[q']$ in terms of moments of the MaxEnt distribution.

### 4.1. From Cardinalities to Probabilities

We start from an observation:

$$\mathbf{E}[|q(\bar{x})|] = \sum_{\bar{c}\in D^t} \Pr[q(\bar{x}/\bar{c})] \text{ where } t = |\bar{x}|$$

where $q(\bar{x}/\bar{c})$ means the Boolean query that results from substituting $x_i$ with $c_i$ for $i = 1, \ldots, t$, where $t$ is the number of head variables in $q$. This is true since,

$$\mathbf{E}[|q(\bar{x})|] = \mathbf{E}[\ |\{\bar{c} \mid q(\bar{x}/\bar{c})\}|\ ] = \sum_{\bar{c}\in D^t} \mathbf{E}[|q(\bar{x}/\bar{c})|] = \sum_{\bar{c}\in D^t} \Pr[q(\bar{x}/\bar{c})]$$

The first equality is the definition of $|q(\bar{x})|$, the second is linearity of expectation. The last is that the expectation of a Boolean random variable is equivalent to its probability.

Let $\bar{v}$ be the views used in a statistical program, $q(\bar{x})$ be the query whose size we want to estimate, and $C$ be the set of all constants mentioned in $\bar{v}$ and $q$. A $C$-permutation of the domain $D$ is a bijection $f : D \to D$ that is invariant on $C$. The MaxEnt model is invariant under $C$-permutations, meaning that for any instance $I$, $\Pr[I] = \Pr[f(I)]$, for any $C$-permutation $f$. Therefore, $\Pr[q(\bar{x}/\bar{c})]$ is the same for all constants $\bar{c} \in D - C$. We exploit this in order to simplify the formula above, as illustrated by this example:

*Example* 4.1. Assume no constants occur in the views $\bar{v}$, and consider the query $q(x, y, z) = R(x, y), R(y, z), x \neq y, y \neq z, x \neq z$. Then:

$$\mathbf{E}[|q(x, y, z)|] = \sum_{c_1,c_2,c_3} \Pr[q(c_1, c_2, c_3)] \ = \ \langle N\rangle_{(3)} \Pr[q(a_1, a_2, a_3)]$$

where $\langle N\rangle_{(k)} = N(N-1)\cdots(N-k+1)$ is the falling factorial. Here $a_1, a_2, a_3$ are three distinct, fixed (but arbitrary) constants, and $q(a_1, a_2, a_3) = R(a_1, a_2), R(a_2, a_3)$. The case without inequalities, $q(x, y, z) = R(x, y), R(y, z)$, can be handled similarly, by considering five cases: (1) $a_1 \neq a_2 \neq a_3 \neq a_1$, (2) $a_1 = a_2 \neq a_3 \neq a_1$, ..., (5) $a_1 = a_2 = a_3$, leading to $\mathbf{E}[|q(x, y, z)|] = \langle N\rangle_{(3)} \Pr[q(a_1, a_2, a_3)] + \langle N\rangle_{(2)} \Pr[q(a_1, a_1, a_3)] + \ldots + \langle N\rangle_{(1)} \Pr[q(a_1, a_1, a_1)]$.

We generalize the example. Let $\bar{x} = \{x_1, \ldots, x_t\}$ be the query's head variables, and let $A = \{a_1, \ldots, a_t\}$ be distinct constants, that do not occur in $C$ (the set of constants in $\bar{v}$ and $q(\bar{x})$). Consider all substitution $\theta : \{x_1, \ldots, x_t\} \to A \cup C$: call $\theta, \theta_1$ equivalent if there exists a $C$-permutation $f$ such that $\theta_1 = f \circ \theta$. We want to retain a single substitution from a set of equivalent substitutions, and for that we retain the smallest one in lexicographic order. If $\theta < \theta_1$ denotes the lexicographic order on substitutions[8], then we call $\theta$ *canonical* if for any other equivalent substitution $\theta_1$, we have $\theta < \theta_1$. Let $\Theta$ be the set of canonical substitutions.

Proposition 4.2. *With the notations above:*

$$\mathbf{E}[|q(\bar{x})|] \ = \ \sum_{\theta\in\Theta} \langle N - |C|\rangle_{(|\theta(\bar{x})\cap A|)} \Pr[q(\theta(\bar{x}))]$$

The proof is straightforward, by a direct extension of the example above.

### 4.2. Probabilities for Simple Programs

A *full query* is a Boolean query without variables, e.g., $q = R(a, b), R(a, d)$ is a full query. We give here an explicit equation for $\Pr_\Sigma[q]$ over the MaxEnt distribution given by a program $\Sigma$, for the case when $\Sigma$ is either a simple hypergraph program, or a simple relational program. Note that, in probabilistic databases [Dalvi and Suciu 2007], computing the probability of $q$ for a full query is

[8]The lexicographic order is defined in a standard fashion. First, define the following order $<$ on $A \cup C$: $a_i < a_j$ for all $1 \leq i < j \leq t$; $a_i < c_k$ forall $i \in [t]$, $k \in [|C|]$; and $c_k < c_l$ forall $1 \leq k < l \leq |C|$. Then the lexicographic order $\theta < \theta_1$ is defined as: $\exists i.\theta(i) < \theta_1(i)$ and $\forall j < i, \theta(i) = \theta_1(i)$.

trivial, because all tuples are assumed to be either independent or factored into independent sets. MaxEnt models, however, are not independent, and cannot be decomposed into simple independent factors. As a result, computing $\Pr_\Sigma[q]$ is non-trivial. Computing $\Pr_\Sigma[q]$ intimately relies on the combinatorics of the underlying MaxEnt distribution, and so, we are only able to compute $\Pr_\Sigma[q]$ directly for some programs.

**Simple Hypergraph Programs** We start with the case of a simple hypergraph program $\Sigma_{Hm}$. Recall that the schema is $S(A_1, \ldots, A_m)$, $R_1(A_1), \ldots, R_m(A_m)$, and $\Sigma_{Hm}$ consists of the constraints $S.A_1 \subseteq R_1, \ldots, S.A_m \subseteq R_m$, and the statistics $\#S = d, \#R_1 = d_1, \ldots, \#R_m = d_m$. The parameters of the MaxEnt model are $\gamma$ (for $\#S$) and $\beta_i$ for $\#R_i$), $i = 1, \ldots, m$.

Let $q = g_1, g_2, \ldots$ be a full conjunctive query, i.e., each $g_i$ is a grounded tuple. Let $I(q)$ be the smallest database instance that satisfies $q$. $I(q)$ can be obtained from $q$ by chasing the inclusion constraints: it consists of all tuples $g_1, g_2, \ldots$ occurring in $q$, and, in addition, of all tuples $R_i(a_i)$ s.t. $q$ contains some atom $S(\bar{a})$ such that $(\bar{a})_i = a_i$. Define:

$$u_i = \left|R_i^{I(q)}\right|, \quad i = 1, \ldots, m \qquad\qquad u_S = \left|S^{I(q)}\right|$$

If $q = S(a,b), S(a,d), R_1(a), R_2(c)$, then $I(q) = \{R_1(a), R_2(b), R_2(c), R_2(d), S(a,b), S(a,d)\}$, $u_1 = 1, u_2 = 3, u_S = 2$.

Denote $\langle X \rangle_{(k)} = X(X-1)\cdots(X-k+1)$, the $k$-falling factorial. Given a probability space Pr, we write $A_i$ for the random variable $|R_i.A_i| = |R_i|$. Then $\mathbf{E}[\langle A_i \rangle_{(u)}]$ denotes the expected value of the $u$-falling factorial of $A_i$; it can be computed directly as $\sum_{\bar{k}} \langle k_i \rangle_{(u)} t(\bar{\alpha}, \gamma, \bar{k})$ in time $O(N^m)$ (see Prop 2.6), and we give more effective methods in the next section.

Theorem 4.3. *With the notation above, let $\Sigma_{Hm}$ be a hypergraph program of arity $m$ over a domain of size $N$. Then, following equation holds:*

$$\Pr_{\Sigma_{Hm}}[q] = \left(\frac{\gamma}{1+\gamma}\right)^{u_S} \left(\prod_{i=1,\ldots,m} \langle N \rangle_{(u_i)}\right)^{-1} \mathbf{E}_{\Sigma_{Hm}}\left[\prod_{i=1,\ldots,m} \langle A_i \rangle_{(u_i)}\right]$$

This theorem allows us to reduce query answering to moment computation. Thus, if we can compute moments of the MaxEnt distribution (and know the parameter $\gamma$), we can estimate query cardinalities. We defer the proof to Section 4.3 and instead inspect some examples and show how we use this formula.

*Example* 4.4. Let $q = S(a,b), S(a,d), R_1(e), R_2(c)$. Then $u_1 = 2$, $u_2 = 3$, $u_S = 2$. We have:

$$\Pr_{\Sigma_{H2}}[q] = \left(\frac{\gamma}{1+\gamma}\right)^2 \frac{\mathbf{E}_{\Sigma_{H2}}[A_1(A_1-1)A_2(A_2-1)(A_2-2)]}{N^2(N-1)^2(N-2)}$$

Here $\mathbf{E}_{\Sigma_{H2}}[A_1(A_1-1)A_2(A_2-1)(A_2-2)]$ denotes the expected value of $|R_1| \cdot (|R_1|-1) \cdot |R_2| \cdot (|R_2|-1) \cdot (|R_2|-2)$.

*Example* 4.5. Given a binary relation $S(A,B)$, define the *fanout* $X_a$ of a constant $a$ is the number of tuples $(a,b) \in S$. Computing the expected fanout is an important problem in query optimization. Fix two constants $b \neq b'$; we have $\mathbf{E}[X_a] = 1 + (N-1)\Pr[S(a,b') \mid S(a,b)]$. Applying Bayes' Rule gives us:

$$\mathbf{E}_{\Sigma_{H2}}[X_a] = 1 + (N-1)\frac{\Pr_{\Sigma_{H2}}[S(a,b), S(a,b')]}{\Pr_{\Sigma_{H2}}[S(a,b)]} = 1 + \frac{\gamma}{1+\gamma}\frac{\mathbf{E}_{\Sigma_{H2}}[A \cdot B \cdot (B-1)]}{\mathbf{E}_{\Sigma_{H2}}[A \cdot B]}$$

Finally, we show how to derive an interesting identity between the expectations of $|S|$ and of the product $\prod_i |R_i|$ for a hypergraph program $\Sigma_{Hm}$:

Corollary 4.6. $\mathbf{E}_{\Sigma_{Hm}}[|S|] = \frac{\gamma}{1+\gamma}\mathbf{E}_{\Sigma_{Hm}}[\prod_{i=1,m} |R_i|]$.

PROOF. We have $\mathbf{E}_{\Sigma_{Hm}}[|S|] = N^m \cdot \Pr_{\Sigma_{Hm}}[S(a_1,\ldots,a_m)] = \frac{\gamma}{1+\gamma}\mathbf{E}_{\Sigma_{Hm}}[A_1 \ldots A_m]$, by applying Theorem 4.3 to the query $q = S(a_1,\ldots,a_m)$; the Corollary follows from the fact that $A_i = |R_i|$. □

Notice that, in the Corollary, $\mathbf{E}_{\Sigma_{Hm}}[|S|]$ is exactly $d$, the statistics on $\#S$ in the program $\Sigma_{Hm}$. However, $\mathbf{E}_{\Sigma_{Hm}}[|R_1|\ldots|R_m|]$ is not easily related to $d_1,\ldots,d_m$; while $\mathbf{E}_{\Sigma_{Hm}}[|R_i|] = d_i$, the random variables $A_1,\ldots,A_m$ are not independent, hence $\mathbf{E}_{\Sigma_{Hm}}[A_1 \ldots A_m]$ is not the product $\prod_{i=1}^m d_i$.

**Simple Relational Programs** Next, we discuss the case when $\Sigma_{Rm}$ is a simple relational program of arity $m$: $R(A_1,\ldots,A_m)$, with statistics $\#R.A_i = d_i$ for $i = 1,\ldots,m$, $\#R = d$, and no constraints. Let $\alpha_i$, $i = 1,\ldots,m$ and $\gamma$ be its parameters. A full query $q$ consists of a set of atoms of the form $R(\bar{c})$. Construct a new hypergraph program $\Sigma_{Hm}$, by normalizing $\Sigma_{Rm}$: it has schema $R(A_1,\ldots,A_m), Q_1(A_1),\ldots,Q_m(A_m)$, constraints $R.A_i \subseteq Q_i$, $i = 1,\ldots,m$, and parameters $\beta_i = \alpha_i/(1-\alpha_i)$, $i = 1,\ldots,m$. The MAXENT distribution given by $\Sigma_{Hm}$ is a probability space with outcomes $R, Q_1,\ldots,Q_m$; from Theorem 3.1 (applied $m$ times) it follows that the marginal distribution of $R$ is precisely the MAXENT distribution for the $\Sigma_{Rm}$-program. This discussion implies:

COROLLARY 4.7. $\Pr_{\Sigma_{Rm}}[q] = \Pr_{\Sigma_{Hm}}[q]$.

*Example* 4.8. Consider a relational program $\Sigma_R$ over a relation $R(A,B)$ given by the statistics $\#R = d$, $\#R.A = d_1$, $\#R.B = d_2$. Suppose we want to estimate the size of the query $q(x,y,z) = R(x,y), R(z,y), x \neq z$. Thus, $\mathbf{E}_{\Sigma_R}[|q|] = N^2(N-1)\Pr_{\Sigma_R}[q(a,b,c)]$, where $a,b,c$ are three fixed constants s.t. $a \neq c$. To compute $\Pr_{\Sigma_R}[q(a,b,c)]$ we first normalize the program (using Theorem 3.1) obtaining the hypergraph program $\Sigma_H$, over relations $R(A,B), R_1(A), R_2(B)$, constraints $R.A \subseteq R_1, R.B \subseteq R_2$, and statistics $\#R = d', \#R_1 = d_1', \#R_2 = d_2'$. Let $\gamma$ be the parameter for $\#R$ in $\Sigma_H$. Then $\Pr_{\Sigma_R}(q(a,b,c)) = \Pr_{\Sigma_H}(q(a,b,c)) = \left(\frac{\gamma}{1+\gamma}\right)^2 \frac{\mathbf{E}_{\Sigma_H}[|R_1|(|R_1|-1)|R_2|]}{N^2(N-1)}$, implying $\mathbf{E}_{\Sigma_R}[|q|] = \left(\frac{\gamma}{1+\gamma}\right)^2 \mathbf{E}_{\Sigma_H}[|R_1|\,(|R_1|-1)\,|R_2|]$.

**General Conjunctive Queries** So far we have considered only queries without existential variables. Consider a Boolean conjunctive query $q$ with $v$ existential variables, $q = \exists\bar{y}.\phi(\bar{y},\bar{c})$, $|\bar{y}| = v$. Then, we can express $\Pr[q]$ in terms of $O(N^v)$ moments. We illustrate here the main idea, on one example, the relational program $\Sigma_R$ in Figure 1: given $\#R, \#R.A, \#R.B, \#R.C$, compute $\mathbf{E}_{\Sigma_R}[|R.AC|]$. We have $\mathbf{E}_{\Sigma_R}[|R.AC|] = N^2\Pr_{\Sigma_R}[q(a,c)]$, where $q(x,z) = \exists y.R(x,y,z)$ and $q(a,c)$ is the Boolean query $q' = \exists y.R(a,y,c)$. After normalizing that relational program, we obtain the hypergraph program $\Sigma_H$ over $R(A,B,C), R_1(A), R_2(B), R_3(C)$ with constraints $R.A \subseteq R_1$, $R.B \subseteq R_2$, $R.C \subseteq R_3$, and we have $\Pr_{\Sigma_R}(q') = \Pr_{\Sigma_H}(q')$. We cannot apply directly Theorem 4.3 because the query $q'$ has an existential variable $y$. Instead, we write $q' \equiv \bigvee_{b\in D} R(a,b,c)$, then apply the inclusion-exclusion formula:

$$\begin{aligned}\Pr_{\Sigma_H}[q'] &= \sum_{B\subseteq D: B=\{b_1,\ldots,b_k\}} (-1)^{k+1}\Pr_{\Sigma_H}[R(a,b_1,c),\ldots,R(a,b_k,c)] \\ &= \sum_{k\geq 1}(-1)^{k+1}\binom{N}{k}\left(\frac{\gamma}{1+\gamma}\right)^k \frac{\mathbf{E}_{\Sigma_H}[A\cdot\langle B\rangle_{(k)}\cdot C]}{N^2\cdot\langle N\rangle_{(k)}}\end{aligned}$$

Each moment above can be computed in time $O(k\cdot N^3)$, and there are $O(N)$ moments to compute. In practice, however, one may stop when $k \ll N$. For example, when computing Figure 1, taking $k = 3$, the error $\varepsilon$ satisfied $|\varepsilon| \leq 10^{-10}$.

### 4.3. Proof Of Theorem 4.3

Fix a hypergraph program $\Sigma_{Hm}$; recall its schema is $S(\bar{A}), R_i(A_i)$, $i = 1,\ldots,m$, and its constraints are $\Gamma = \{S.A_i \subseteq R_i \mid i = 1,m\}$. Let $q = g_1,\ldots,g_s$ be a full query over these relations.

Recall the general form of the partition function, given in Definition 2.2. For hypergraph programs, we have shown in Proposition 2.14 that it admits a simpler form:

$$T = T^{\Sigma_{Hm}}(\bar{\alpha}; \gamma) = \sum_{W \in \mathsf{I}(\Gamma)} \prod_{i=1,\ldots,m} \alpha_i^{|R_i^W|} \cdot \gamma^{|S^W|} = \sum_{\bar{k}} (1+\gamma)^{\prod_{i=1,\ldots,m} k_i} \times \prod_{i=1,\ldots,m} \binom{N}{k_i} \alpha_i^{k_i}$$

Denote by $T_q$ the partition function over the constraints $\Gamma \cup \{q\}$; in other words, it sums only over the worlds $W$ such that $W \models q$, and, thus, $\Pr_{\Sigma_{Hm}}[q] = \frac{T_q}{T}$. A world $W$ satisfies $q$ iff it contains $I(q)$, because the latter is the smallest database instance that satisfies $q$. Using the same argument as in the proof of Proposition 2.14, we have:

$$T_q = \sum_{W \in \mathsf{I}(\Gamma): W \supseteq I(q)} \prod_{i=1,\ldots,m} \alpha_i^{|R_i^W|} \cdot \gamma^{|S^W|} = \underbrace{\left(\gamma^{u_S} \times \prod_{i=1,\ldots,m} \alpha^{u_i}\right)}_{(\dagger)} \times \sum_{\bar{k}} \left( \prod_{i=1,\ldots,m} \binom{N-u_i}{k_i} \alpha_i^{k_i} \right) \times (1+\gamma)^{\prod_{i=1,\ldots,m}(k_i+u_i)-u_S}$$

The term (†) counts tuples in $I(q)$; the remaining part of the expression counts the other tuples. The expression simplifies to:

$$\begin{aligned} T_q &= \left(\frac{\gamma}{1+\gamma}\right)^{u_S} \prod_{i=1,\ldots,m} \alpha^{u_i} \times \sum_{\bar{k}} \left( \prod_{i=1,\ldots,m} \binom{N-u_i}{k_i} \alpha_i^{k_i} \right) \times (1+\gamma)^{\prod_{i=1,\ldots,m}(k_i+u_i)} \\ &= \left(\frac{\gamma}{1+\gamma}\right)^{u_S} \sum_{\bar{k}} \left( \prod_{i=1,\ldots,m} \binom{N-u_i}{k_i-u_i} \alpha_i^{k_i} \right) \times (1+\gamma)^{\prod_{i=1,\ldots,m} k_i} \end{aligned}$$

The first line is algebra. The second line is simply renumbering (and observing that $\binom{N}{k} = 0$ for $k < 0$). We claim that:

$$\frac{T_q}{T} = \left(\frac{\gamma}{1+\gamma}\right)^{u_S} \mathbf{E}\left[ \prod_{i=1,\ldots,m} \frac{\langle A_i \rangle_{(k_i)}}{\langle N \rangle_{(k_i)}} \right]$$

which immediately implies the theorem, because $\Pr_{\Sigma_{Hm}}[q] = \frac{T_q}{T}$. We prove the claim:

$$\begin{aligned} \left(\frac{\gamma}{1+\gamma}\right)^{u_S} \mathbf{E}\left[ \prod_{i=1,\ldots,m} \frac{\langle A_i \rangle_{(u_i)}}{\langle N \rangle_{(u_i)}} \right] &= \left(\frac{\gamma}{1+\gamma}\right)^{u_S} \frac{1}{T} \prod_{i=1,\ldots,m} \frac{1}{\langle N \rangle_{(u_i)}} \alpha_i^{u_i} \frac{\partial}{\partial \alpha_i} T \\ &= \left(\frac{\gamma}{1+\gamma}\right)^{u_S} \frac{1}{T} \sum_{\bar{k}} \prod_{i=1,\ldots,m} \binom{N}{k_i} \frac{\langle k_i \rangle_{(u_i)}}{\langle N \rangle_{(u_i)}} \alpha^{k_i} \times (1+\gamma)^{\prod_{i=1,\ldots,m} k_i} \\ &= \left(\frac{\gamma}{1+\gamma}\right)^{u_S} \frac{1}{T} \sum_{\bar{k}} \prod_{i=1,\ldots,m} \binom{N-u_i}{k_i-u_i} \alpha^{k_i} \times (1+\gamma)^{\prod_{i=1,\ldots,m} k_i} = \frac{T_q}{T} \end{aligned}$$

The last line uses the binomial identity:

$$\binom{N}{k} \frac{\langle k \rangle_{(v)}}{\langle N \rangle_{(v)}} = \frac{\langle N \rangle_{(k)}}{k!} \frac{\langle k \rangle_{(v)}}{\langle N \rangle_{(v)}} = \frac{\langle N-v \rangle_{(k-v)}}{(k_i - v)!} = \binom{N-v}{k-v}$$

□

## 5. MODEL COMPUTATION

We first describe the solutions for chain programs. These programs can be solved in closed form. Then, we discuss the Peaks Approximation technique, which is used to solve the model computation problem for hypergraphs and binary relational programs.

### 5.1. Warm up: Chain Programs

Consider a chain program of size $m$, $\Sigma_{Cm} = (\Gamma, \bar{v}, \bar{d})$. For an example when $m = 2$, see Example 2.13. Recall that the partition function for a chain program is defined by the recurrence (Proposition 2.12):

$$T^{C0}() = 1 \text{ and } T^{Cj+1}(\alpha_1, \ldots, \alpha_{j+1}) = \left(1 + \alpha_{j+1} T^{Cj}(\alpha_1, \ldots, \alpha_j)\right)^N$$

We show that we can write a simple equation for the moments of the chain program:

Proposition 5.1. *Given a chain program $\Sigma_{Cm}$ of size m, then for $j = 1, \ldots, m$*

$$\mathbf{E}[|R_j|] = \prod_{i=j,\ldots,m} N \frac{\alpha_i T^{Ci-1}}{1 + \alpha_i T^{Ci-1}}$$

Proof. This follows directly from the following calculation:

$$\frac{\partial}{\partial \alpha_j} T^{Ci} = \begin{cases} T^{Ci} \times N \frac{T^{Ci-1}}{1+\alpha_i T^{Ci-1}} & \text{if } j = i \\ T^{Ci} \times N \frac{\alpha_i}{1+\alpha_i T^{Ci-1}} \frac{\partial}{\partial \alpha_j} T^{Ci-1} & \text{if } j < i \end{cases}$$

Then, we use this inductively:

$$\alpha_j \frac{\partial}{\partial \alpha_j} T^{Cm} = T^{Cm} \times \prod_{i=j,\ldots,m} N \frac{\alpha_i T^{Ci-1}}{1 + \alpha_i T^{Ci-1}}$$

We conclude by observing that $\alpha_j \frac{\partial}{\partial \alpha_j} T^{Cm} = T^{Cm} \times \mathbf{E}[|R_j|]$. □

We now give an $O(m)$ time algorithm to solve the model computation problem by observing the following identity:

$$\frac{d_j}{d_{j+1}} = \frac{\mathbf{E}[|R_j|]}{\mathbf{E}[|R_{j+1}|]} = N \frac{\alpha_j T^{Cj-1}}{1 + \alpha_j T^{Cj-1}}$$

The recursive procedure starts with $T^{C0} = 1$ in the base case; recursively, we compute the value $T^{Ci}$ and all moments. We observe that this uses no asymptotic approximations. Summarizing, we have shown:

Theorem 5.2. *Given a chain program $\Sigma$ of arity m the above algorithm solves the model computation problem in time $O(m)$ for any domain size.*

### 5.2. Overview of The Peaks Approximation

The Peaks Approximation writes a MaxEnt distribution as a convex sum of simpler distributions using two key pieces of intuition. First, in many cases, almost all of the mass in the partition function comes from a small fraction of its terms. Second, around each peak, the function behaves like a simpler function (here, a product of binomials).

To make this intuition more concrete, consider the following hypergraph program $\Sigma_{H2}$: $\#R_1.A_1 = 2$, $\#R_2.A_2 = 4$ and $\#S = 10$ on a domain of size $N = 99$. In Figure 2, we solve the model and then plot $\ln f(k, l)$ where $f(k, l) = \max\{t^{\Sigma_{H2}}(k, l), e^{-10}\}$ and $t^{\Sigma_{H2}}(k_1, k_2)$ is the associated term function: $k_1$ is on the $x$ axis, and $k_2$ is on the $y$ axis, and on the $z$-axis is $\ln t(x, y)$. Most of the mass of $t^{\Sigma_{H2}}(k, l)$ is concentrated around $t(2, 4)$, i.e., around the expected values given in the program, and some slightly smaller mass is concentrated around $t(99, 99)$. The idea of the Peaks Approximation is to locally approximate the term function $t$ in the neighborhood of $(2, 4)$ and $(99, 99)$ with simpler functions.

The formal setting that we consider in this section is as follows: we are given a simple hypergraph program $\Sigma_{Hm}$ of size $m$ with relations $R_1 \ldots, R_m$ and $S$. We are also given $\bar{\alpha} = \alpha_1, \ldots, \alpha_m$ and

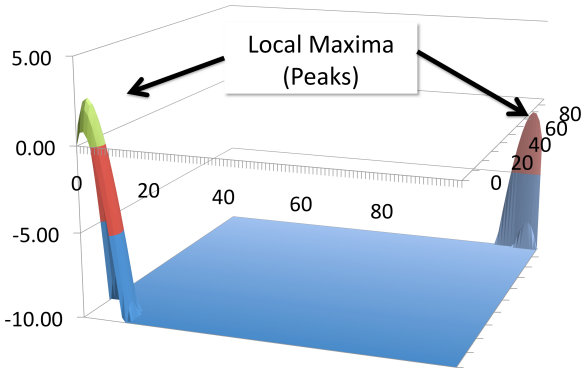


Fig. 2. A graph of $\ln t(k,l)$ for the hypergraph program with $\#R.A = 2$ $\#R.B = 4$), $\#R = 10$ and $N = 99$. For readability, we plot $\ln f(k,l)$ where $f(k,l) = \max\{t(k,l), e^{-10}\}$. Almost all mass comes from the two peaks.

$\gamma$ the parameters of the MaxEnt distribution associated with $\Sigma_H$. Intuitively, we want to approximate the MaxEnt distribution with a convex sum of products of binomials. Formally, we devise an approximation for $t^{\Sigma_H}$, the term function associated with $\Sigma_H$.

We describe the Peaks Approximation in three steps. (1) We define the functions that we use in the Peaks Approximation, and (2) we describe how to find the parameters of the approximation that we define in step (1). Finally, in step (3) we give a technical lemma that defines a sufficient condition for the Peaks Approximation to be a good approximation.

*Step (1): The Approximating Functions.* Our approximation function will be a weighted sum of two products of binomials: it is parametrized by two tuples of values $\bar{c}^{(1)}, \bar{c}^{(2)} \in \mathbb{R}^m$ that represent the center of each product of binomials.[9] Let Peaks $= \left\{\bar{c}^{(1)}, \bar{c}^{(2)}\right\}$. In the next section, we show how to find Peaks. For now, we define a function $\tilde{t}(\bar{\alpha}, \gamma; \bar{k})$ that approximates $t^{\Sigma_{Hm}}$.

$$\tilde{t}(\bar{\alpha}, \gamma; \bar{k}) = \sum_{\bar{c} \in \text{Peaks}} (1+\gamma)^{(1-m)P(\bar{c})} \times \prod_{i=1}^{m} \binom{N}{k_i} \alpha_i^{k_i} (1+\gamma)^{k_i/c_i P(\bar{c})} \text{ where } P(\bar{c}) = \prod_{i=1}^{m} c_i$$

The partition function associated to the Peaks Approximation, $\tilde{T}$ is obtained by summing $\tilde{t}$ over $\bar{k}$:

$$\tilde{T}(\bar{\alpha}, \gamma) = \sum_{\bar{c} \in \text{Peaks}} (1+\gamma)^{(1-m)P(\bar{c})} \times \prod_{i=1,\ldots,m} \left(1 + \alpha_i (1+\gamma)^{\frac{k_i}{c_i} P(\bar{c})}\right)^N \tag{7}$$

We can view the Peaks Approximation as replacing the complicated MaxEnt distribution $T$ with the simpler function $\tilde{T}$. In the next section, we show how to find Peaks and so specify $\tilde{T}$.

*Step (2): Finding the set* Peaks. Fix a hypergraph program $\Sigma$. We take Peaks to be the set of local maxima for the term function $t^{\Sigma_{Hm}}$. Intuitively, this is where $T^{\Sigma}$'s mass is concentrated, so it makes sense to locally approximate $t$ near the peaks. Below, we show a surprising fact: for hypergraph programs, there are at most two local maxima (justifying our notation above).

Theorem 5.3 (Number of Peaks). *Let $\Sigma_{Hm}$ be a simple hypergraph program of size $m$. Let $t^{\Sigma_H}$ be the term function of $\Sigma_{Hm}$. Then, for any fixed $(\bar{\alpha}, \gamma)$ such that $\alpha_i > 0$ $i = 1, \ldots, m$, $\gamma > 0$, $t^{\Sigma H}(\bar{\alpha}, \gamma; \bar{k})$ has at most 2 local maxima in $\bar{k}$.*

We prove this theorem in Section 5.4 in several steps. The first observation is that we can interpolate $t$ with a smooth function (i.e., a continuously differentiable function). Second, we observe that

[9] It is straightforward to extend this idea to a mixture of more than a product of two binomials (i.e. |Peaks| $\geq 2$). We do not need this generalization for our theoretical development in this paper.

a local maxima of $t(\bar{\alpha}, \gamma; \bar{k})$ function must be a critical point.[10] Then, we observe that, by the *mean value theorem* [Rudin 1976, pg. 108], to find a critical point it suffices to find values of $\bar{k}$ such that $t(\bar{k}) = t(\bar{k} + e^{(i)})$ for $i = 1, \ldots, m$ where $e^{(i)}$ is the unit vector in direction $i$]. This process yields a system of equations (one equation for each $k_1, \ldots, k_m$). We then show that all the solutions of this system of equations are the zeros of a function of a single variable. In turn, we show that the now one-dimensional function has at most 3 zeros by showing that the third derivative of this function has a constant sign. Then, we conclude that at most 2 critical points can be local maxima.

*Step (3): Sufficient Conditions for Approximation.* Informally, we give a sufficient condition about the set Peaks that allows us to conclude the Peaks Approximation is a good approximation to the hypergraph partition function. The lemma is unfortunately technical and requires three conditions, which intuitively say: (1) that the error around each peak is small enough, (2) the peaks are far enough apart, and (3) that the peaks are not in the middle of the space. Given these conditions, we show that the Peaks Approximation is asymptotically strong: every finite moment of the Peaks Approximation is an asymptotic approximation of the original distribution.[11]

LEMMA 5.4. *Fix a hypergraph program $\Sigma_{Hm}$. Let $N = 1, 2, \ldots$, and let $T_N^{\Sigma}$ denote the partition function for $\Sigma$ on a domain of size $N$. For each $N$, let $\tilde{T}_N$ be the Peaks Approximation for $T_N^{\Sigma}$ and $c_i^{(l,N)}$ for $l = 1, 2$ and $i = 1, \ldots, m$ denote the local maxima of $t^N$. We make the following three assumptions: (1) $\ln(1+\gamma)N^{m-2} = o(1)$, (2) $\min_{i=1,\ldots,m} \left|c_i^{(1,N)} - c_i^{(2,N)}\right| \geq N^{-\varepsilon}$ for some $\varepsilon > 0$, and (3) for $l = 1, 2$ $\exists i$ such that $\min\left\{c_i^{(l,N)}, N - c_i^{(l,N)}\right\} = O(N^{1-\tau})$ for some $\tau > 0$. With these assumptions, for any tuple $\bar{s}$ of $m$ positive numbers:*

$$\lim_{N\to\infty} \frac{\mathbf{E}_{T_N}[\prod_{i=1,\ldots,m}\langle A_i\rangle_{(s_i)}]}{\mathbf{E}_{\tilde{T}_N}[\prod_{i=1,\ldots,m}\langle A_i\rangle_{(s_i)}]} = 1$$

We prove this lemma in Section 5.5 by showing two statements. The first statement informally says that the peaks are a best local, linear approximation (in the exponent), and we use this to write the error of the Peaks Approximation in a closed form. The second result is a variation of the standard Chernoff Bound [Mitzenmacher and Upfal 2005], which is the typical tool used to say that random binomial variables are very sharply (exponentially) concentrated about their mean. The proof of Lemma 5.4 then boils down to a calculation that combines these two statements.

*Moment Computation for the Peaks Approximation.* We give a closed-form solution for moments of the Peaks Approximation as a weighted sum of each peak:

THEOREM 5.5. *Let $\tilde{T}$ be a Peaks Approximation (Eq. 7) defined by* Peaks *with parameters $\alpha_1, \ldots, \alpha_m, \gamma$. Then, for any $\bar{s} \in \mathbb{N}^m$ the following equation holds:*

$$\mathbf{E}\left[\prod_{i=1,\ldots,m}\langle A_i\rangle_{(s_i)}\right] = \sum_{\bar{c}\in\text{Peaks}} w(\bar{c}) \prod_{i=1,\ldots,m} N\frac{\alpha_i^{s_i}(1+\gamma)^{s_i P(\bar{c})/c_i}}{1+\alpha_i^{s_i}(1+\gamma)^{s_i P(\bar{c})/c_i}}$$

*where $N$ is the size of the domain and*

$$w(\bar{c}) = \left(\tilde{T}(\bar{\alpha}, \gamma)\right)^{-1} \sum_{\bar{k}} (1+\gamma)^{(1-m)P(\bar{c})} \times \prod_{i=1}^{m} \binom{N}{k_i} \alpha_i^{k_i} (1+\gamma)^{k_i/c_i P(\bar{c})}$$

Notice that $w(\bar{c}) \geq 0$ for $\bar{c} \in$ Peaks and $\sum_{\bar{c}\in\text{Peaks}} w(\bar{c}) = 1$. Coupled with the fact that each term in the summation is the moment of a binomial, this justifies our statement that the Peaks Approxima-

[10] Given a continuously differentiable function $f : \mathbb{R}^m \to \mathbb{R}$, a *critical point* is a point $x \in \mathbb{R}^m$ such that $\partial x_i t(x) = 0$ for $i = 1, \ldots, m$.

[11] To be clear about the order of quantifiers: the value of $\bar{s}$ is chosen, and then the limit $N$ is applied in the theorem.

tion is a mixture of binomials. Combining Theorem 5.5 with Theorem 4.3, we can approximate any full query in $O(|q|)$-time using the Peaks Approximation.

Next, we use this sufficient condition to verify asymptotic solutions for several statistical programs.

### 5.3. Asymptotic Model Computation Solutions

We state the asymptotic solutions for simple hypergraph programs and simple binary (arity 2) relational programs.

*Hypergraph Programs.* We solve hypergraph programs of any arity.

THEOREM 5.6. *Consider a hypergraph programs of arity $m \geq 2$, where (without loss) $0 < d_1 \leq d_2 \leq \cdots \leq d_m < d_S = O(1)$ then the following parameters are an asymptotic solution:*

$$\alpha_i = d_i N^{-1} \text{ and } \gamma = gN^{1-m} + N^{-m}\left(\delta + \ln \frac{d_S}{Ng}\right)$$

*where $g = -\sum_{i=1,\ldots,m} \ln \frac{\alpha_i}{1+\alpha_i}$, and we set $\delta = g^2/2 - (d_1 + d_2)$ if $m = 2$ and $\delta = 0$ if $m > 2$.*

The strange looking $\delta$ term for $m = 2$ arises for a technical reason: we need to balance a limit appropriately. The key to the proof of Theorem 5.6 is to establish following lemma that describes the local maxima of $t^{\Sigma_{HM}}$ above parameters.

LEMMA 5.7. *With the parameters and notation of Theorem 5.6, the set of local maxima for $t^{\Sigma_{Hm}}$ are $\left\{\bar{d} + \delta^{(1)}, \bar{c}^{(2)} + \delta^{(2)}\right\}$ where $\bar{c}^{(2)} = (N - d_2, N - d_1)$ if $m = 2$ and $\bar{c}^{(2)} = (N, \ldots, N)$ otherwise; and $\bar{\delta}^{(i)}$ is a vector such that $\max_j |\delta_j| = O(N^{-1})$. Moreover, in the notation of Theorem 5.5, $w(\bar{c}^{(2)}) = \frac{d_S}{Ng}$ and $w(\bar{c}^{(1)}) = 1 - w(\bar{c}^{(2)})$.*

Observe that the conditions of Lemma 5.4 are satisfied, so that we may use the peaks instead of the MAXENT to calculate the moments. Using the fact that $w(\bar{c}^{(2)}) = o(N^{-1})$ it is straightforward to calculate the moments of the distribution and verify that $\mathbf{E}[|R_i|] = d_i \cdot w(c^{(1)}) + N \cdot w(c^{(2)}) = d_i + o(1)$ and $\mathbf{E}[|S|] = 0 \cdot w(c^{(1)}) + N^m \frac{\gamma}{1+\gamma} \cdot w_2 = d_S \frac{1}{1+\gamma} \to d_S$. Anecdotally, we have implemented this statistical program and verified that the values converge within small errors for small $N$ (on the order of hundreds) for a broad range of values of $\bar{d}$. We return to the proof of this Lemma in Appendix B.

*Binary Relations.* Our solution for binary relations combines normalization and the Peaks approach, but there is a subtle twist. Recall the binary relational program $\Sigma_{R2}$ is over a binary relation $R(A, B)$ with assertions $\#R.A = d_A$, $\#R.B = d_B$, and $\#R = d_R$. If we try to directly reuse the solutions from Theorem 5.6 for $\Sigma_{H2}$, and we set the hypergraph parameters to any constant, then the normalization tells us that both $|R.A|$ and $|R.B|$ tend to zero with increasing $N$, i.e.,

$$\mathbf{E}_{\Sigma_{R2}}[|R.A|] = (1 + \alpha_1)\mathbf{E}_{\Sigma_{H2}}[|R_1|] - N\alpha_1 \approx d_i - d_i \to 0$$

It turns out, finding the solution for binary relations require subtle balancing:

THEOREM 5.8. *Given $\Sigma_{R2}$ above assume that $d_A \leq d_B \leq d_R$. Then, the tuple of parameters $(\alpha_1, \alpha_2, \gamma)$ defined as follows is an asymptotic solution for $\Sigma_{R2}$: Let $\frac{\alpha_1}{1+\alpha_1} = aN^{-1}$, $\frac{\alpha_2}{1+\alpha_2} = bg_1^{-1}$ and $\gamma = g_1 N^{-1} + g_2 N^{-2}$ where*

$$a = (d_A + 1)/(e^b - 1),\ b = d_b/d_a \text{ and } g_1 = -\mathsf{W}_{-1}(-\alpha b)$$

$$g_2 = g_1^2/2 + (1 + \beta)\ln(1 + \beta)\frac{d_G - d_B}{N \ln g_1}$$

*Here, $\mathsf{W}_{-1}$ denotes the value of the Lambert W function over the non-principal (but real-valued) branch.*

The proof uses normalization to transform the program into a hypergraph program, and then use the Peaks Approximation instead of the MaxEnt distribution (via Lemma 5.4). We include these calculations in Appendix C along with a proof of the above.

We solve programs with non-binary relations using numeric techniques based on solving the set of equations defined by Equation 8.

### 5.4. Proof of Theorem 5.3

Fix $m \geq 2$. Given a hypergraph program $\Sigma_{Hm}$ and values for $N, \bar{\alpha}, \gamma$, our goal is to find (and characterize) the set of local maxima for $t^{\Sigma_{Hm}}$. The technical problem is that the set of maxima is only known through a system of equations – and this system of equations has many variables. This implies that the solution set could be infinite. We show, however, that the solution set is finite, by showing that all the solutions must lie along some curve in 1d, i.e., we show there is a function $f$ with domain $\mathbb{R}$ that characterizes all solutions. Then, we find the roots of this one dimensional equation. More strongly, we show that there are at most two maxima, proving the theorem.

We begin with an observation. Let $\bar{1}$ denote the vector $(1, \ldots, 1) \in \mathbb{R}^m$. Suppose that the term function has a (local) maximum value at some point $\bar{k} \in \mathbb{N}^m$. Then, we have the following pair of of inequalities:

$$t(\bar{k} - \bar{1}) \leq t(\bar{k}) \text{ and } t(\bar{k} + \bar{1}) \leq t(\bar{k})$$

Now, $t$ can be viewed as a continuous function with type $\mathbb{R}^m \to \mathbb{R}$. From this fact, we can deduce that there must exist some $\bar{l} \in \mathbb{R}^m$ such that $\max_{i=1,\ldots,m} |l_i - k_i| \leq 1$ and $t(\bar{l}) = t(\bar{l} + \bar{1})$. To see why, consider the function $g(s) = t(\bar{l} + s\bar{1}) - t(\bar{l} + (1 + s)\bar{l})$. The inequalities above suggest that $g(0) \geq 0$ while $g(1) \leq 0$. Hence, there is some $s \in [0, 1]$ such that $g(s) = 0$, which implies our above statement. Exactly symmetric reasoning applies for minima. Thus, we can find all local maxima and minima by solving the following system of equations:

$$t(\bar{k}) = t(\bar{k} - e_i) \text{ for } i = 1, \ldots, m$$

We denote by $K_\Pi = \prod_{i=1}^m k_i$. Then the resulting system of equations to describe a peak is equivalent to the following using the identity $\binom{N}{k_i}\binom{N}{k_i - 1}^{-1} = \frac{N+1-k_i}{k_i}$ (for real-valued $k_i$).

$$\frac{N + 1 - k_i}{k_i} \alpha_i (1 + \gamma)^{K_\Pi / k_i} = 1 \text{ for } i = 1, \ldots, m \tag{8}$$

Fix some $i$, $\alpha_i$, and $\gamma$, then the above equations define a pair of functions $(f_i, g_i)$ for $i = 1, \ldots, N$ where $g_i, f_i : \mathbb{R} \to \mathbb{R}$. Let $f_i$ be the function that maps $K_\Pi$ to $k_i$ and let $g_i$ denote its inverse. We show that these functions can be used to characterize the solutions of the equations. We observe that $g_i$ can be found with straightforward arithmetic

$$g_i(k_i) = -\frac{1}{\ln(1 + \gamma)} \left( \ln \frac{N + 1 - k_i}{k_i} + \ln \alpha_i \right)$$

Computing, $f_i$, however, requires more work and the use of a special function, Lambert's $\mathsf{W}$ (multi)function, which is defined by $\mathsf{W}(u) = v$ means that $ve^v = u$. We derive an explicit form for $f_i$ in terms of $\mathsf{W}$:

Lemma 5.9. *Let $f_i$ be defined as above, then for each $i = 1, \ldots, m$ $f_i$ can be written explicitly as:*

$$f_i(K_\Pi) = (N + 1) \left( 1 + \frac{1}{cK_\Pi} \mathsf{W}\left( cK_\Pi e^{-cK_\Pi} \frac{1}{\alpha_i} \right) \right)^{-1}$$

*where $c = \frac{1}{N+1} \ln(1 + \gamma)$, i.e., $c$ does not depend on $K_\Pi$.*

Proof of Lemma 5.9. We need to do some algebra to get the solution into the form where W can help us. Define $c = N^{-1}(\ln(1+\gamma))$ and let $x_i = (N+1)k_i$ thus our equation becomes:

$$\frac{1-x_i}{x_i}\alpha_i(1+\gamma)^{(N+1)^{-1}K_\Pi/x_i} = 1$$

we rearrange terms:

$$-\left(1-\frac{1}{x_i}\right)\exp\left\{cK_\Pi\frac{1}{x_i}\right\} = \frac{1}{\alpha}$$

Now write $z_i = 1 - \frac{1}{x_i}$ (so that $\frac{1}{x_i} = 1 - z_i$). Using this substitution and rewriting we have:

$$z_i \exp\{-cK_\Pi z_i\} = -\frac{1}{\alpha}e^{-cK_\Pi}$$

And now, let $v_i = -cK_\Pi z_i$ so that $z_i = -\frac{v_i}{cK_\Pi}$ leaving:

$$v\exp\{v\} = cK_\Pi e^{-ck_\Pi}\frac{1}{\alpha}$$

Inverting this equation using the $W$ function gives:

$$v = W\left(\frac{1}{\alpha}cK_\Pi e^{-cK_\Pi}\right) \implies z = -\frac{1}{cK_\Pi}\mathsf{W}\left(\frac{1}{\alpha}cK_\Pi e^{-cK_\Pi}\right) \implies x = \frac{1}{1+\frac{1}{cK_\Pi}W(\frac{1}{\alpha}cK_\Pi e^{-cK_\Pi})}$$

□

Any critical point is a zero of the following equation:

$$\Phi(K_\Pi) = \prod_{i=1,\ldots,n} f_i(K_\Pi) - K_\Pi$$

We show that $\Phi$ has at most 3 zeros, and, so at most two can be local maxima. Thus, Figure 2 represents the picture of the general case for hypergraph programs.

Lemma 5.10. *Assuming that $\alpha_i > 0$ for $i = 1, \ldots, m$, $\Phi(K_\Pi)$ has at most 3 zeros.*

Intuitively, we show that the logarithmic derivative is concave, which implies the original function has at most 3 zeros using the *mean value theorem* [Rudin 1976, p.108].

Proof. We first convert this equation into an equivalent form using log, here we use that $K_\Pi > 0$.

$$\sum_{i=1,\ldots,m} \log f_i(K_\Pi) = \log K_\Pi$$

The following calculations are easiest to verify with the computer algebra system, Maple. We differentiate this equation with respect to $K_\Pi$. We use the fact that:

$$\frac{d\mathsf{W}(x)}{dx} = \frac{W(x)}{x(1+W(x))}$$

$$\sum_{i=1,\ldots,m} \frac{\mathsf{W}(gK_\Pi e^{-gK_\Pi}\alpha_i^{-1})}{K_\Pi(1+\mathsf{W}(gK_\Pi e^{gK_\Pi}\alpha_i^{-1}))} = \frac{1}{K_\Pi}$$

Which in turn reduces to:

$$-\sum_{i=1,\ldots,m} \frac{\mathsf{W}(gK_\Pi e^{-gK_\Pi}\alpha_i^{-1})}{(1+\mathsf{W}(gK_\Pi e^{gK_\Pi}\alpha_i^{-1}))} = 1$$

Now, we show that this equation has at most 2 solutions, by showing that its derivative has 1 solution. Taking a derivative, we have:

$$(1 - gK_\Pi) \sum_{i=1,t} \frac{\mathsf{W}(gK_\Pi e^{-gK_\Pi}\alpha_i^{-1})}{K_\Pi(1 + \mathsf{W}(gK_\Pi e^{-gK_\Pi}\alpha_i^{-1}))^3} = 0$$

We observe that the summation is always positive, since $\alpha_i > 0$. In turn, in this range moreover $\mathsf{W}$ is a single-valued, positive function. Thus, there is exactly one zero: when $g = \frac{1}{K_\Pi}$, hence the second derivative has exactly one zero, proving the claim. □

We observe the main theorem as a corollary. Since $T$ is the partition function for simple hypergraphs, this immediately implies Theorem 5.3 holds.

### 5.5. Proof of Lemma 5.4

We now state a proposition that precisely spells out the local, relative error of using the Peaks Approximation to approximate a hypergraph program; this will allow us to provide sufficient conditions for $\tilde{T}$ to be a good approximation (defined formally below).

Proposition 5.11. *Let $\Sigma$ be a hypergraph of arity $m$ and let $\bar{c}$ be a tuple of constants of arity $m$. Let $t = t^\Sigma$ be the term function and then let $\tilde{t}(\bar{\alpha}; \bar{k}, \gamma; \bar{c})$ be a (single peak) approximation. Further, let $\Theta$ be any derivative (operator) generated by composing finitely many partial derivatives from the set $\left\{\frac{\partial}{\partial \alpha_i}\right\}_{i=1,\ldots,m}$. Then, for any $\bar{\delta} \in Z^m$ we have:*

$$\frac{\Theta[t(\bar{c} + \bar{\delta})]}{\Theta[\tilde{t}(\bar{c} + \bar{\delta})]} = \exp\left\{\ln(1+\gamma)P(\bar{c})\left(\prod_{i=1,\ldots,m}\left(1 + \frac{\delta_i}{c_i}\right) - 1 - \sum_{i=1,\ldots,m}\frac{\delta_i}{c_i}\right)\right\} \tag{9}$$

*except if $\Theta[t(\bar{c} + \bar{\delta})] = 0$ in which case $\Theta[\tilde{t}(\bar{c} + \bar{\delta})] = 0$ as well.*

Deriving this equation is straightforward, but we can read some interesting things from it: First, the Peaks Approximations is preserved under taking moments. Intuitively, assuming the Peaks Approximation is good, we can use the approximation to compute moments. A second point is that $\bar{c}$ is an arbitrary point in the statement above, i.e., $\bar{c}$ is not necessarily a local maximum of $T$, and so, our approximation is in some sense a best local, linear approximation (in the exponent) for $T$ about $\bar{c}$.

A second key fact that we need is about the constituent parts of $\tilde{T}$: binomials. A binomial distributed random variable is tightly concentrated (e.g., Chernoff's bound [Alon and Spencer 1992, p. 270]). Translating this fact into our notation is the following result that most of the mass of the binomial is on a small number of terms:

$$\sum_{k:|y-\mu|\geq\delta\mu}\binom{N}{k}\alpha^k \leq 2(1+\alpha)^N \exp\left\{-\delta^2\mu/2\right\} \text{ where } \mu = N\frac{\alpha}{1+\alpha}$$

We need a generalization of this observation around each peak. For $\bar{\delta} \in \mathbb{R}_+^m$ and a point $\bar{\mu} \in \mathbb{R}^m$, define the *$\bar{\delta}$-neighborhood* around $\bar{\mu}$ as the following set:

$$\text{Nbd}(\bar{\mu}, \bar{\delta}) = \{\bar{k} \in \mathbb{N}^m \mid |\mu_i - k_i| \leq \delta_i \text{ for } i = 1, \ldots, m\}$$

Also, we say that $f(N) = \tilde{O}(g(N))$ if $f = O(g(N)\text{polylog}(N))$.

Lemma 5.12. *With the notation of the theorem. We assume that $\ln(1+\gamma) = \tilde{O}(N^{1-m})$ and that there exists some $\tau > 0$ and some $i$ such that $\min\{c_i, N - c_i\} = O(N^{1-\tau})$. Let $\bar{c} \in \left\{\bar{c}^{(1)}, \bar{c}^{(2)}\right\}$. Then, for any $s > 0$, there exists a tuple $\bar{\delta} \in \mathbb{R}^m$ that satisfies the following two conditions simultaneously.*

*(1) For each $i = 1, \ldots, m$. $N^{-s} \geq q_i$ where $q_i = \min\left\{\exp\left\{-c^i\delta_i^2/2\right\}, \exp\left\{-(N - c_i)\delta_i^2/2\right\}\right\}$*

(2) *For any* $\bar{y} \in \text{Nbd}(\bar{c}, \bar{\delta})$ *such that* $|y_i - c_i| \leq \delta_i$, *the relative error at* $\bar{y}$ *(Eq. 9) tends to* 0 *as* $N \to \infty$.

Proof Lemma 5.12. We set $\delta_i$ so that the first equation holds with equality and satisfies condition (1) i.e., $\delta_i = \sqrt{2s}c_i^{-1/2} \ln N$. It suffices to prove the condition on the border, so consider some $\bar{y} = \bar{c} \pm \bar{\delta}$. To simplify notation, write $\delta_i = zc_i^{-1/2}$ where $z = \sqrt{2s \ln N}$. The following chain of inequalities establishes (2). We can write the exponent of Eq. 9 as (Let $C$ be some constant)

$$\begin{aligned}
\ln(1+\gamma)P(\bar{c}) \sum_{X \subseteq \{1,\ldots,m\}: |X| \geq 2} \prod_{i \in X} \frac{\delta_i}{c_i} &= \ln(1+\gamma) \sum_{X \subseteq \{1,\ldots,m\}: |X| \geq 2} z^{|X|} \left( \prod_{i \in X} c_i^{1/2} \right) \left( \prod_{j \in \neg X} c_j \right) \\
&\leq C \ln(1+\gamma) \sum_{X \subseteq \{1,\ldots,m\}: |X| \geq 2} z^{|X|} \left( N^{|X|/2(1-\tau)} \right) \left( N^{|\neg X|(1-\tau)} \right) \\
&= C \binom{m}{2} \ln(1+\gamma)(N^{m-1-\tau} + O(N^{m-3/2-\tau}) \ln^m N \to 0
\end{aligned}$$

The first inequality is the assumption that $c_i = O(N^{1-\tau})$ which implies there is some $C$. The second inequality observes that the high order term is when $|X| = 2$. Thus, since $\ln(1+\gamma) \ln^m N = \tilde{O}(N^{1-m})$ then $\ln(1+\gamma)N^{m-1-\tau} \ln^m N = o(1)$ which is less than any $\varepsilon$, proving the claim. □

We now prove Lemma 5.4. Property (2) says that for any $\varepsilon > 0$ and large enough $N$, the relative error of all terms in the $\bar{\delta}$ neighborhood is less than $\varepsilon$. Moreover, the terms on the the frontier of this neighborhood contribute $O(N^{-m})$. Then, we observe a following simple fact about the Peaks Approximation: *every term not contained in one of these balls must be smaller than some term on the frontier of one of these neighborhoods*. To see this consider the following graph: each term in the partition function is a node and it adds a directed edge to its highest valued neighbor. Now, the only sinks (with out degree 0) in this graph are local maxima. And so, any node $\bar{v}$ has a path from itself to one local maxima, call it $\bar{k}_0$. Either the node is within the neighborhood, or the path must cross the frontier of the neighborhood $\bar{k}_0$, and so, the value $t(\bar{v})t(\bar{k}_0)^{-1} = o(N^{-m})$. Since there are at most $N^m$ such terms taking $k > m$ suffices to show the asymptotic statements. Moreover, since the terms are $N^{-\Omega(\varepsilon)}$ apart the two binomials contribute only a negligible amount inside each others $\bar{\delta}$ neighborhood.

## 6. EXTENSION: BUCKETIZATION

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 $\text{P}^{\leq}$ the set of project queries with range predicates. We 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. 6.1 we have $v_1(x, y) :- R(x, y), x < 0.25N$. We leave non-asymptotic assertions such as $R.A < 10$ for future work.

*Example* 6.1. **Overlapping Ranges** Consider two views[12]:

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

and the statistical program $\#v_1 = d_1, \#v_2 = d_2$. 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 MaxEnt model gives us a precise distribution that represents only these observations and nothing more. The partition function is $(1 + x_1)^{N_1}(1 + x_1x_2)^{N_2}(1 + x_2)^{N_3}$, and the MaxEnt distribution has the form $\Pr[I] = \omega\alpha_1^{k_1}\alpha_2^{k_2}$, where $k_1 = |I \cap (D_1 \cup D_2)|$ and $k_2 = |I \cap (D_2 \cup D_3)|$; we show below how to compute the parameters $\alpha_1, \alpha_2$.

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

Let $\bar{R} = R_1, \ldots, R_m$ be a relational schema, and consider a statistical program $\Sigma, \Gamma$ with range queries, over the schema $\bar{R}$. We translate it into a *bucketized* statistical program $\Sigma^0, \Gamma^0$, 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 $\Sigma$ the following holds

$$\sum_{u \in \texttt{buckets}(v)} c_u = d \tag{10}$$

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

Then the following holds:

Proposition 6.2. *Let $\Sigma^0, \Gamma^0$ be the bucketized program for $\Sigma, \Gamma$. Let $\bar{\beta} = (\beta_k)$ be the* MaxEnt *model of the bucketized program. Consider some parameters $\bar{\alpha} = (\alpha_j)$. Suppose that for every statistical assertion $\#v_j = d_j$ in $\Sigma$ condition (10) holds, and the following condition holds for every query $u_k \in BV$:*

$$\beta_k = \prod_{j : u_k \in \texttt{buckets}(v_j)} \alpha_j \tag{11}$$

*Then $\bar{\alpha}$ is a solution to the* MaxEnt *model for $\Sigma, \Gamma$.*

This gives us a general procedure for solving the MaxEnt model for programs with range predicates: introduce new unknowns $c_j^{\bar{i}}$ and add Equations (10) and (11), then solve the MaxEnt model for the bucketized program under these new constraints.

*Example* 6.3. Recall Example 6.1: we have two statistics $\#\sigma_{A \le 0.60N}(R) = d_1$, and $\#\sigma_{A \ge 0.25N}(R) = d_2$. 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 (11):

$$\begin{aligned} \beta_1 &= \alpha_1 \\ \beta_2 &= \alpha_1\alpha_2 \\ \beta_3 &= \alpha_2 \end{aligned}$$

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 (10), $c^1 + c^2 = d_1$ and $c^2 + c^3 = d_2$ to obtain:

$$\begin{aligned} N_1 \frac{\alpha_1}{1+\alpha_1} + N_2 \frac{\alpha_1\alpha_2}{1+\alpha_1\alpha_2} &= d_1 \\ N_3 \frac{\alpha_2}{1+\alpha_2} + N_2 \frac{\alpha_1\alpha_2}{1+\alpha_1\alpha_2} &= d_2 \end{aligned}$$

Solving this gives us the MaxEnt model. Consistent histograms [Srivastava et al. 2006] had a similar goal of using MaxEnt to capture statistics on overlapping intervals, but use a different, simpler probabilistic model based on frequencies.

## 7. RELATED WORK

The first body of related work is in cardinality estimation. As noted above, while a variety of synopses structures have been proposed for cardinality estimation [Ioannidis 2003; Olken 1993; Deligiannakis et al. 2007; Alon et al. 1999], 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 [Ioannidis and Christodoulakis 1991]). In contrast, we ask the question: *given some statistical information, what is the best estimate that one can make?*

The MaxEnt model has been applied in prior work to the problem of cardinality estimation [Markl et al. 2005; Srivastava et al. 2006]. However, the focus was restricted to queries that consist of conjunctive selection predicates over single tables. In contrast, we explore a full-fledged MaxEnt model that can incorporate statistics involving arbitrary first-order expressions. There are more technical differences as well: the previous model applies the MaxEnt model to the space of frequencies of observed statistics, which are continuous real-valued observables. It is non-trivial to use MaxEnt in continuous settings (as it may no longer uniquely defined) [Jaynes 2003, p.377]. In contrast, we consider the MaxEnt to the probability distribution on the underlying (discrete) relations. In this case, the estimates MaxEnt provides are unique. That said, MaxEnt is still only a principle, and other principles are possible [Kass and Wasserman 1996]. Additionally, our technical results differ from the above approach in two ways: (1) we show how to predict full conjunctive queries using the MaxEnt approach (in contrast, prior work focused on single-table histogram estimates), and (2) we find asymptotic solutions to the MaxEnt models (in contrast, prior work used numerical techniques).

In our previous work [Kaushik et al. 2009], we introduced the MaxEnt model over possible worlds for computing statistics, and solved it in a very limited setting, when the MaxEnt distribution is a random graph. We left open the MaxEnt models for cardinality estimation that are not random graphs, such as the models we solve in this paper. In another work [Kaushik and Suciu 2009], we discussed a MaxEnt model for set/bag semantics: we did not discuss bag semantics in this paper. Also prior art did not address query estimation. Entropy maximization is a well-established principle in statistics for handling incomplete information [Jaynes 2003].

The MaxEnt principle also underlies the graphical model approach, notably the *probabilistic relational model* [Getoor et al. 2001] and *Markov Logic Networks* [Domingos and Richardson 2004]. In particular both approaches require that the underlying probability distribution is in the exponential family (which is dictated by an MaxEnt approach). Recall that the classical central limit theorem (CLT) is an asymptotic statement that provides conditions under which the mean of a sample of large enough number of random variables will be approximately normally distributed. Our asymptotic results in this paper are inspired by such results, and we can view our results as a first step toward a kind of central limit theorem for graphical models. In the same way that the CLT provided a basis for many independent sampling related tasks, such a theory would allow for such tasks where the structure of the task was specified by graphical models. For example, our results may provide an initial estimate for the optimization programs that underlie parameter estimation in graphical models. It is interesting future work to explore how the techniques in this paper apply to inference and learning in such approaches, e.g., *Factor Graphs* [Sen and Deshpande 2007] and *Markov Logic Networks* [Richardson and Domingos 2006].

Probabilistic databases [Dalvi and Suciu 2007; Antova et al. 2007; Koch and Olteanu 2008; Widom 2005] focus on efficient query evaluation over a probabilistic database, in which probabilities are specifies with tuples. Our focus is on computing the parameters of a different type of models.

## 8. CONCLUSION AND FUTURE WORK

We propose to model database statistics using maximum entropy probability distributions. This model is attractive because any query has a well defined size estimate, all statistics act as a whole, and the model extends smoothly when new statistics are added. As part of our technical development we described three techniques: normalization, query answering via moments, and the Peaks Approximation that we believe are of both theoretical and practical interest for solving statistical programs.

The next step for our work is to implement a prototype cardinality estimator using the theoretical underpinnings laid out in this paper.

Our work raises several theoretical directions. One direction is to find a general theory of numerical solutions for richer MaxEnt models. While an analytic solution is the gold standard for MaxEnt models, many applications of cardinality estimation can tolerate approximate solutions. In our experiments for this paper, we solved many of models (many more than we can solve theoretically) using numerical techniques. For even moderate domain sizes, applying the direct entropy equation is infeasible (as there is one variable for each of the exponentially many possible worlds). Empirically, we have had some success solving the peak equations numerically. However, determining the right mathematical optimization approach to solve MaxEnt models in general is an intriguing open question.

A second direction is to understand the complexity of decision procedures that a general cardinality estimator built on the MaxEnt theory must support. For example, deciding whether or not an arbitrary statistical program is affinely independent or is satisfiable for logical models for large enough $N$ is an open question. In contrast, there is an obvious but potentially exponential decision procedure for each $N$. In this work, we have seen that for simple models we can decide these properties easily – but we do not have a general procedure for these problems. Efficient special cases of such procedures could have practical applications: they would enable optimizers to find contradictions in their statistical specifications (say as a result of estimates that have changed due to updates, inserts, and deletes of the underlying databases).

## ACKNOWLEDGMENTS

The authors would like to thank Raghav Kaushik for thoughtful discussion and collaboration on the early stages of this work and the reviewers for their careful reading of this manuscript and for their detailed feedback. The authors would also like to thank Ce Zhang for his help preparing the graphics in this manuscript.

## REFERENCES

Abiteboul, S., Hull, R., and Vianu, V. 1995. *Foundations of Databases*. Addison Wesley Publishing Co.

Alon, N., Gibbons, P. B., Matias, Y., and Szegedy, M. 1999. Tracking join and self-join sizes in limited storage. In *PODS*. 10–20.

Alon, N., Matias, Y., and Szegedy, M. 1996. The space complexity of approximating the frequency moments. In *STOC*. 20–29.

Alon, N. and Spencer, J. 1992. *The Probabilistic Method*. John Wiley.

Antova, L., Koch, C., and Olteanu, D. 2007. World-set decompositions: Expressiveness and efficient algorithms. In *ICDT*. 194–208.

Boyd, S. and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

Chaudhuri, S., Narasayya, V. R., and Ramamurthy, R. 2008. Diagnosing estimation errors in page counts using execution feedback. In *ICDE*. 1013–1022.

Corless, R. M., Jeffrey, D. J., and Knuth, D. E. 1997. A sequence of series for the lambert w function. In *ISSAC*. 197–204.

Dalvi, N. N., Miklau, G., and Suciu, D. 2005. Asymptotic conditional probabilities for conjunctive queries. In *ICDT*. 289–305.

Dalvi, N. N. and Suciu, D. 2007. The dichotomy of conjunctive queries on probabilistic structures. In *PODS*. 293–302.

Deligiannakis, A., Garofalakis, M. N., and Roussopoulos, N. 2007. Extended wavelets for multiple measures. *ACM Trans. Database Syst. 32,* 2, 10.

Domingos, P. and Richardson, M. 2004. Markov logic: A unifying framework for statistical relational learning. In *ICML Workshop on Statistical Relational Learning*. 49–54.

Getoor, L., Taskar, B., and Koller, D. 2001. Selectivity estimation using probabilistic models. In *SIGMOD Conference*. 461–472.

Haas, P. J., Naughton, J. F., Seshadri, S., and Swami, A. N. 1996. Selectivity and cost estimation for joins based on random sampling. *J. Comput. Syst. Sci. 52,* 3, 550–569.

Ioannidis, Y. E. 2003. The history of histograms (abridged). In *VLDB*. 19–30.

Ioannidis, Y. E. and Christodoulakis, S. 1991. On the propagation of errors in the size of join results. In *SIGMOD Conference*. 268–277.

Jaynes, E. 2003. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, UK.

Kass, R. E. and Wasserman, L. 1996. The selection of prior distributions by formal rules. *Journal of the American Statistical Association 91,* 435, pp. 1343–1370.

Kaushik, R., Ré, C., and Suciu, D. 2009. General database statistics using entropy maximization. In *DBPL*. 84–99.

Kaushik, R. and Suciu, D. 2009. Consistent histograms in the presence of distinct value counts. *PVLDB 2,* 1, 850–861.

Koch, C. and Olteanu, D. 2008. Conditioning probabilistic databases. *PVLDB 1,* 1, 313–325.

Markl, V., Megiddo, N., Kutsch, M., Tran, T. M., Haas, P. J., and Srivastava, U. 2005. Consistently estimating the selectivity of conjuncts of predicates. In *VLDB*. 373–384.

Mitzenmacher, M. and Upfal, E. 2005. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA.

Olken, F. 1993. Random sampling from databases. Ph.D. thesis, University of California at Berkeley.

Papadimitriou, C. 1994. *Computational Complexity*. Addison Wesley Publishing Company.

Poosala, V. and Ioannidis, Y. E. 1997. Selectivity estimation without the attribute value independence assumption. In *VLDB*. 486–495.

Ré, C. and Suciu, D. 2010. Understanding cardinality estimation using entropy maximization. In *PODS*. 53–64.

Richardson, M. and Domingos, P. 2006. Markov logic networks. *Machine Learning 62,* 1-2, 107–136.

Rudin, W. 1976. *Principles of Mathematical Analysis, Third Edition* 3rd Ed. McGraw-Hill Science/Engineering/Math.

Rusu, F. and Dobra, A. 2008. Sketches for size of join estimation. *ACM Trans. Database Syst. 33,* 3.

Sen, P. and Deshpande, A. 2007. Representing and querying correlated tuples in probabilistic databases. In *ICDE*. 596–605.

Shao, J. 2003. *Mathematical Statistics* 2nd Ed. Springer.

Srivastava, U., Haas, P. J., Markl, V., Kutsch, M., and Tran, T. M. 2006. Isomer: Consistent histogram construction using query feedback. In *ICDE*. 39.

Stillger, M., Lohman, G. M., Markl, V., and Kandil, M. 2001. Leo - db2's learning optimizer. In *VLDB*. 19–28.

Wainwright, M. J. and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning 1,* 1-2, 1–305.

Widom, J. 2005. Trio: A system for integrated management of data, accuracy, and lineage. In *CIDR*. 262–276.

# Online Appendix to: Understanding Cardinality Estimation using Entropy Maximization

CHRISTOPHER RÉ, University of Wisconsin–Madison
DAN SUCIU, University of Washington, Seattle

## A. PROOF OF THEOREM 2.4

In this section, we reprove some folklore statements that we were unable to track down proofs; these results are not contributions of this work (for a variant of these results see Wainwright and Jordan [Wainwright and Jordan 2008, §3.2]).

Fix a domain size $N$. Given a program $\Sigma = (\Gamma, \bar{v}, \bar{d})$ over a space of instances $\mathsf{I}(\Gamma)$, Let $P(\Gamma)$ denote all probability distributions over $\mathsf{I}(\Gamma)$. The set $P(\Gamma)$ is a closed, bounded subset of $\mathbb{R}^{|\mathsf{I}(\Gamma)|}$, thus it is compact. Moreover, $P(\Gamma)$ is convex.

We say that $\Sigma$ is *satisfiable* if there exists $\bar{p} \in P(\Gamma)$ such that $F(\bar{p}) = \bar{d}$. A hypergraph program $\Sigma_H = (\bar{v}, \bar{d})$ is consistent over a domain of size $N$ if $\bar{d}$ is in the convex hull of the vectors: $(\bar{c}, z)$ where $z = \prod_{i=1,\ldots,m} c_i$ where $\bar{c} \in \{0, \ldots, N\}^m$.

Given a set of views $\bar{v}$ define $E : P \to \mathbb{R}^t$ by $E(\bar{p}) = \bar{c}$ where

$$\bar{c}_j = \sum_{I \in \text{Inst}} \bar{p}_I \left| v_j(I) \right|$$

Let $H$ denote the entropy, i.e., $H(\bar{p}) = -\sum_{I \in \text{Inst}} \bar{p}_I \log \bar{p}_I$. $H$ is a continuous, real-valued function. Moreover $-H(\bar{p})$ is a convex function since its Hessian is only non-zero on the diagonal, $\frac{\partial}{\partial p_I}^2 - H(\bar{p}) = p_I^{-1}$ and all other (mixed) second derivatives are 0. This shows that $-H$ it is positive definite on the interior of $P(\Gamma)$, which is equivalent to convexity [Boyd and Vandenberghe 2004, pg. 65].

### A.1. Maximum Entropy Distribution Exists

Proposition A.1. *The set $E^{-1}(\bar{d})$ is compact.*

Proof. We observe that $E$ is continuous. Hence, $E^{-1}(\bar{d})$ is a closed set. Since $P(\Gamma)$ is compact, this means that $E^{-1}(\bar{d})$ is a closed subset of a compact set, and so compact. □

Thus, the entropy $H$ takes a maximum value on the set. Formally,

$$\sup_{\bar{p} \in E^{-1}(d)} H(\bar{p}) = H(\bar{q})$$

for some $\bar{q} \in E^{-1}(\bar{d})$, which proves that there is at least one maximum entropy probability distribution.

### A.2. Uniqueness

Proposition A.2. *Given a satisfiable statistical program $\Sigma$, then there is a unique probability distribution that satisfies $\Sigma$.*

Proof. Consider the negative entropy function $-H(\bar{p})$. By compactness and continuity of $-H$, $-H(\bar{p})$ attains a minimum value on $P(\Gamma)$ provided $P(\Gamma)$ is not empty (which since $\Sigma$ is satisfiable it is not). By convexity of $P(\Gamma)$ and strict convexity of $-H(\bar{p})$, there is a single point that obtains a minimum value. Thus, there is a unique minimal value of the negative entropy, and hence a single distribution with maximum entropy. □

© YYYY ACM 0362-5915/YYYY/01-ARTA $10.00
DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

Given a set of $|\bar{v}|$ parameters, $\bar{\alpha}$, let $P$ be the function that maps $\bar{\alpha}$ to a probability distributions $p_\alpha$ over $\mathsf{I}(\Gamma)$ defined by

$$p_\alpha(I) = \frac{1}{Z} \prod_{i=1,\ldots,m} \alpha_i^{|v_i(I)|} \text{ where } Z = \sum_{J \in \mathsf{I}(\Gamma)} \prod_{i=1,\ldots,m} \alpha_i^{|v_i(J)|}$$

We now give a sufficient condition for $P$ to be injective. We say that a set of views $\bar{v}$ where $|\bar{v}| = m$ is *affinely dependent* over $\mathsf{I}(\Gamma)$ if there exist real numbers $\bar{c}$ and a value $d$ such that (1) $c_i$ are not all zero and (2) the following holds:

$$\forall I \in \mathsf{I}(\Gamma). \quad \sum_{j=1,\ldots,m} |v_j(I)| c_j = d$$

If no such $(\bar{c}, d)$ exists, we say that the views are *affinely independent*.

Proposition A.3. *Fix a set* $\mathsf{I}(\Gamma)$*. If* $\bar{v}$ *is affinely independent over* $\mathsf{I}(\Gamma)$ *then, P mapping* $\alpha$ *to* $p_\alpha$ *is injective.*

Proof. Suppose not, then there exists $\bar{\alpha}, \bar{\beta}$ such that $P(\bar{\alpha}) = P(\bar{\beta})$. This implies that for each $I$, $\log p_\alpha(I) - \log p_\beta(I) = 0$ so that:

$$\log(Z) - \log(Z') = \sum_{j=1,\ldots,m} |v_j(I)|(\log \alpha_j - \log \beta_j)$$

But then, define $c_j = \log \alpha_j - \log \beta_j$ and $d = \log(Z) - \log(Z')$, then $(\bar{c}, d)$ is a tuple of constants violating the affine independence condition, a contradiction. □

Now we are ready to show:

Theorem A.4. *If* $\Sigma = (\Gamma, \bar{v}, \bar{d})$ *and* $\bar{v}$ *is affinely independent over* $I(\Gamma)$ *and* $\Sigma$ *is satisfiable then there is a unique solution* $\bar{\alpha}$ *that maximizes entropy.*

Proof. Suppose not, then there are two solutions and both are of the form $P(\bar{\alpha})$ and $P(\bar{\beta})$, but this means that $P(\bar{\alpha}) = P(\bar{\beta})$ by Prop A.2. On the other hand, since $\bar{v}$ is affinely independent (by assumption) we have that $P$ is injective (Prop A.3), and so $\bar{\alpha} = \bar{\beta}$, a contradiction. □

*Remark* A.5. The reverse direction of Prop A.3 holds. Thus, this gives a condition to check for a program.

#### *A.2.1. Chains, Hypergraphs, and Relations are Linearly Independent*

Proposition A.6. *A set of vectors is* $\left\{\boldsymbol{x}^{(i)}\right\}_{i=1,\ldots,m}$ *is affinely independent over* $\mathbb{R}^N$ *if and only if* $\left\{\boldsymbol{y}^{(j)}\right\}_{j=1,\ldots,m}$ *where* $y^{(j)} = (x^{(j)}, 1)$ *is linearly independent over* $\mathbb{R}^{N+1}$.

Fix a tuple of views $\bar{v}$. Denote by $\tau_{\bar{v}} : I \to \mathbb{N}^{m+1}$ as $\tau(I) = \bar{t}$ where $t_i = |v_i(I)|$ for $i = 1, \ldots, m$ and $\tau_{m+1} = 1$. We denote the unit vector in direction $i$ as $e^{(i)}$.

Proposition A.7. *A chain program* $\Sigma$ *of size* $m \geq 2$ *is affinely independent for domain sizes* $N \geq 1$.

Proof. Let $I_k = \{R_1(\bar{a}), \ldots, R_i(\bar{a})\}$ so that $\tau(I_k) = \boldsymbol{x}^{(k)}$ where $\boldsymbol{x}_j^{(k)} = 1$ if $j = \{1, \ldots, k\} \cup \{m+1\}$ and $\boldsymbol{x}_j^{(k)} = 0$ otherwise. The set $\{x^{(k)}\}_{k=0,m}$ is a set of $m+1$ linearly independent vectors. □

Proposition A.8. *A hypergraph program of size* $m - 1$ *where* $m \geq 2$ *is affinely independent for for any* $\mathsf{I}(\Gamma)$ *where the domain size is* $N \geq 1$.

Proof. Let $I_i = \{R_i(a)\}$ then $\tau(I_i) = e^{(i)} + e^{(m+2)}$ and $I_{m+1} = \{R_i(a), S(\bar{a})\}$ then $\tau(I_i) = \mathbf{1}$ which is linearly independent. Moreover, $\tau(\emptyset) = e^{(m+1)}$. It is straightforward that this is a linearly independent set. □

PROPOSITION A.9. *A relational program of size $m-1$ where $m \geq 2$ is affinely independent over domains of size $N \geq 2$.*

PROOF. The vectors are $x^{(i)} = \mathbf{1} + e^{(i)} + e^{(m+1)}$ for $i = 1, \ldots, m-1$ (a world with two tuples that differ on one attribute) and $x^{(m)} = \mathbf{1}$ (a world with one tuple) and $x^{(m+1)} = e^{(m+1)}$ (the empty world). □

## B. HYPERGRAPH SOLUTIONS: PROOF OF LEMMA 5.7

We begin with Equation 8 and verify that our claim of where the solutions lie. Technically, our first claim is that there exists $\delta_i$ (which may depend on $N$) such that:

$$\lim_{N\to\infty} \frac{N-(d_i+\delta_i)}{d_i+\delta_i}\alpha_i(1+\gamma)^{\prod_{j:i\neq j} d_j} = 1$$

Since $d_i$ are constant and using the definition of $\alpha_i$, we take any $\delta_i$ we may write:

$$\lim_{N\to\infty} \left(1 - \frac{\delta_i}{d_i+\delta_i} - d_i N^{-1}\right)\left(1+o(N^{-1})\right) = 1 + O(N^{-1}) \tag{12}$$

Notice that we can take $\delta_i = o(N^{-1})$. The upper peak is verified by the equation:

$$\lim_{N\to\infty} \frac{\delta_i}{N-\delta_i}\alpha_i(1+\gamma)^{N^{m-1}} = 1$$

We observe that $\alpha_i(1+\gamma)^{N^m} = \alpha_i\left(\prod_{i=1,\ldots,m}(1+\alpha_i)\alpha^{-1} + \tilde{O}(1)\right) = O(N^m)$. These conditions are first order so the difference between $m=2$ and $m \geq 3$ is not visible. Finally, straightforward (but routine) calculations now show that these are indeed maxima. To see this, observe that Equation 12 is (locally) monontone increasing in the sense that increasing $-\delta_i$ increases the value of each equation. Since this holds for each $\delta_i$ we see that this is a maximum value.

## C. SIMPLE RELATIONAL SOLUTIONS: PROOF OF THEOREM 5.8

We first verify the peak equations for Theorem 5.8. $(d_A, N\frac{\beta}{1+\beta} + d_B)$ is a peak. To see this, for the first component of the first peak, we use the expansion:

$$(1+\gamma)^{N\frac{\beta}{1+\beta}+d_B} = \exp\{b + d_B\gamma + o(1)\}$$

and, $\exp\{b+o(1)\} = O(1)$.

$$\frac{N\alpha(1+\gamma)^{N\frac{\beta}{1+\beta}+d_B} - 1}{1+\alpha(1+\gamma)^{N\frac{\beta}{1+\beta}+d_B}} = \frac{a\exp\{b+o(1)\}-1}{1+O(N^{-1})} = ae^b - 1 + o(1) \to d_A$$

Second, we use the expansion:

$$(1+\gamma)^{d_A} = 1 + \gamma d_A + o(N^{-1})$$

$$\frac{N\beta(1+\gamma)^{d_A}-1}{1+\beta(1+\gamma)^{d_A}} = \frac{N\beta\left(1+\gamma d_A + o(N^{-1})-1\right)-1}{1+\beta(1+\gamma d_A + o(N^{-1}))} = N\frac{\beta}{1+\beta} + bd_A - 1 + o(1) = N\frac{\beta}{1+\beta} + d_B + o(1)$$

To see that the second peak is $(N - (N\frac{\beta}{1+\beta} + d_B), N - d_A)$.

$$\begin{aligned}
\frac{N\alpha(1+\gamma)^{N-d_A}-1}{1+\alpha(1+\gamma)^{N-d_A}} &= \frac{N}{1+(1+\gamma)^{d_A}\beta} - o(1)\\
&= N - N\frac{\beta(1+\gamma)^{d_A}}{1+\beta(1+\gamma)^{d_A}} - o(1)\\
&= N - N\frac{\beta(1+\gamma d_A + o(N^{-1}))}{1+\beta(1+\gamma)^{d_A}} - o(1)\\
&= N - N\frac{\beta}{1+\beta} - d_B
\end{aligned}$$

$$\begin{aligned}
\frac{N\beta(1+\gamma)^{N-(n\frac{\beta}{1+\beta}+d_B)}-1}{1+\beta(1+\gamma)^{(n\frac{\beta}{1+\beta}+d_B)}} &= \frac{N}{1+\alpha(1+\gamma)^{(n\frac{\beta}{1+\beta}+d_B)}} - o(1)\\
&= N - N\frac{\alpha(1+\gamma)^{(n\frac{\beta}{1+\beta}+d_B)}-1}{1+\alpha(1+\gamma)^{(n\frac{\beta}{1+\beta}+d_B)}} = N - d_A
\end{aligned}$$

We now compute the ratios of the two peaks in the approximations: Write $x_i = \alpha(1+\gamma)^{l_i}$ and $y_i = \beta(1+\gamma)^{k_i}$ then,

$$\begin{aligned}
\frac{T_0}{T_N} &= \frac{(1+x_0)^N(1+y_0)^N\left(\frac{\alpha}{x_0}\right)^{k_0}\left(\frac{\beta}{y_0}\right)^{l_0}(1+\gamma)^{k_0 l_0}}{(1+x_1)^N(1+y_1)^N\left(\frac{\alpha}{x_1}\right)^{k_1}\left(\frac{\beta}{y_1}\right)^{l_1}(1+\gamma)^{k_1 l_1}}\\
&= \frac{(1+x_0)^N(1+y_0)^N\left(\frac{\alpha}{x_0}\right)^{k_0}\left(\frac{\beta}{y_0}\right)^{l_0}(1+\gamma)^{k_0 l_0}}{(1+\frac{1}{y_0})^N(1+\frac{1}{x_0})^N y_0^{N-l_0}\alpha^{N-l_0}x_0^{N-k_0}\beta^{N-k_0}(1+\gamma)^{N^2-(k_0+l_0)n+k_0 l_0}}\\
&= \frac{\left(\frac{\alpha}{x_0}\right)^{k_0}\left(\frac{\beta}{y_0}\right)^{l_0}}{y_0^{-l_0}\alpha^{N-l_0}x_0^{-k_0}\beta^{N-k_0}(1+\gamma)^{N^2-(k_0+l_0)n}} g^{-(1-\frac{l_0}{N})} \approx g^{-\frac{1}{1+\beta}}
\end{aligned}$$

Here we have written $\gamma = \frac{\ln\alpha_\pi}{N} + \frac{1}{N^2}\left(\ln g + \frac{\ln^2\alpha}{2}\right)$ so that $(1+\gamma)^{N^2} = (\alpha\beta)^{-N}g$ and $(1+\gamma)^N = (\alpha\beta)^{-1}$. The limit $(1+\gamma)^{l_0 n} = (\alpha\beta)^{l_0} g^{\frac{l_0}{N}}$ since $nl_0$ is close (enough) to $n^2$. With this we can write equation for $T$ (and their intended limits):

$$\mathbf{E}[A] = k_0 + \frac{N-l_0}{g} \to k_0 \qquad \mathbf{E}[B] = l_0 + \frac{N-k_0}{g} \to l_0$$

$$\mathbf{E}[G] = \frac{\gamma}{1+\gamma}\left(k_0 l_0 + g(N-k_0)(N-l_0)\right) \qquad = \mathbf{E}[A]\frac{d_B}{\mathbf{E}[A]} + d_G - d_B = d_G$$

Now, to achieve this we set $g^{\frac{1}{1+\beta}} = \frac{d_g - d_b}{N\ln\alpha_\pi}$. Now, for $S$:

$$\begin{aligned}
\mathbf{E}_S[A] &= k_0(1+\alpha) - n\alpha = a(e^b - 1) - 1\\
\mathbf{E}_S[B] &= l_0(1+\beta) - n\beta = n\beta k_0\gamma = bk_0
\end{aligned}$$

Thus, we take $b = \frac{d_B}{d_A}$ and $a = \frac{d_A+1}{e^b-1}$ and $\gamma$ as above. This is then a solution to $R(A,B)$.