# An Upper Bound on the Global Optimum in Parameter Estimation

**Khaled S. Refaat** and **Adnan Darwiche**
Computer Science Department
University of California, Los Angeles
{krefaat,darwiche}@cs.ucla.edu

## Abstract

Learning graphical model parameters from incomplete data is a non-convex optimization problem. Iterative algorithms, such as Expectation Maximization (EM), can be used to get a local optimum solution. However, little is known about the quality of the learned local optimum, compared to the unknown global optimum. We exploit variables that are always observed in the dataset to get an upper bound on the global optimum which can give insight into the quality of the parameters learned by estimation algorithms.

## 1 Introduction

Probabilistic graphical models (PGMs) have been useful to many fields, including computer vision, bioinformatics, natural language processing, and statistical physics; see [20, 32, 17, 22]. A graphical model represents a joint probability distribution compactly using a structure populated with parameters. In this paper, we consider two types of graphical models: Markov Random Fields (MRFs) and Bayesian networks (BNs).

An MRF consists of an undirected graph defining conditional independence relationships between variables, and a factor for every maximal clique in the graph; see [15, 16, 24]. A Bayesian network consists of a directed acyclic graph associated with conditional probability tables; see [4].

Learning graphical model parameters from data is typically reduced to finding the maximum likelihood parameters: ones that maximize the probability of a dataset, due to their attractive statistical properties [6]. However, due to the complexity of learning maximum likelihood parameters, other simplified methods have also been proposed in literature such as pseudo-likelihood [2], ratio matching [10], composite maximum likelihood [30], contrastive divergence [9], and more recently the LAP algorithm [23].

A key distinction is commonly drawn between complete and incomplete datasets. In a complete dataset, the value of each variable is known in every example in the dataset, whereas in an incomplete dataset, some variables may have missing values. Computationally, learning from incomplete data can be much harder than learning from complete data, as we discuss next.

When the data is complete, learning maximum likelihood parameters can be done efficiently in BNs by one pass through the dataset, and by solving a convex optimization problem in MRFs. However, in MRFs, evaluating the objective or computing the gradient requires doing inference, to compute the partition function, which is #P-hard [27]. Iterative algorithms, such as gradient descent [28], conjugate gradient (CG) [8], L-BFGS [21], iterative proportional fitting (IPF) [13], and more recently EDML [25] can be used to get the global optimum solution.

On the other hand, if the data is incomplete, the optimization problem is generally non-convex, i.e. has multiple local optima. Iterative algorithms, such as expectation maximization (EM) [5, 18] and gradient descent can be used to get a local optimum solution; see Chapter 19 in [24]. The fixed points of these algorithms correspond to the stationary points of the likelihood function. Hence, these algorithms are not guaranteed to converge to global optima. As such, they are typically applied to multiple seeds (initial parameter estimates), while retaining the best estimates obtained across all seeds. However, little is known about the quality of the learned estimates, compared to the unknown global optimum.

In this paper, we propose an upper bound on the unknown global optimum that can give insight into the quality of the learned estimates, as compared to the global optimum. It may also help derive branch-and-bound methods to get the global optimum. Our proposed technique exploits variables that are always observed and requires solving a convex optimization problem. In case of BNs, this convex optimization problem can be solved efficiently.

The paper is organized as follows. In Section 2, we define our notation and give an introduction to the problem of learning graphical model parameters. We propose the MRF and BN upper bounds in Sections 3, and 4, respectively. The experimental results are given in Section 5. We review some of the related work in Section 6, and conclude in Section 7.

## 2 Learning Parameters

In this section, we define our notation, and review how parameter estimation for graphical models is formulated as an optimization problem.

### 2.1 Notation

Upper case letters ($X$) denote variables and lower case letters ($x$) denote their values. Variable sets are denoted by bold-face upper case letters ($\mathbf{X}$) and their instantiations by bold-face lower case letters ($\mathbf{x}$).

We use $\theta$ to denote the set of all network parameters. Parameter learning in graphical models is the process of estimating these parameters $\theta$ from a given dataset.

A *dataset* is a multi-set of *examples.* Each example is an instantiation of some network variables. We will use $\mathcal{D}$ to denote a dataset and $\mathbf{d}_1, \ldots, \mathbf{d}_N$ to denote its $N$ examples. The following is a dataset over four binary variables:

| example | $E$ | $B$ | $A$ | $C$ |
|---|---|---|---|---|
| 1 | $e$ | $b$ | $a$ | ? |
| 2 | ? | $\overline{b}$ | $\overline{a}$ | ? |
| 3 | $e$ | $\overline{b}$ | $\overline{a}$ | ? |

This dataset has three examples, $\mathbf{d}_1$, $\mathbf{d}_2$ and $\mathbf{d}_3$. For a binary variable $X$, we will use $x$ and $\overline{x}$ to denote its two values. Moreover, a "?" indicates a missing value of a variable in an example. The first example above corresponds to instantiation $e, b, a$, while the second example corresponds to instantiation $\overline{b}, \overline{a}$.

A variable $X$ is *fully observed* in a dataset iff the value of $X$ is known in each example of the dataset (i.e., "?" cannot appear in the column corresponding to variable $X$). Variables $A$ and $B$ are fully observed in the above dataset. Moreover, a variable $X$ is *hidden* in a dataset iff its value is unknown in every example of the dataset (i.e., only "?" appears in the column of variable $X$). Variable $C$ is hidden in the above dataset. When all variables are fully observed in a dataset, the dataset is said to be *complete.* Otherwise, the dataset is *incomplete.* The above dataset is incomplete. Finally, we will use $\mathcal{D}_{\mathbf{O}}$ to denote the dataset which results from removing variables outside $\mathbf{O}$ from dataset $\mathcal{D}$.

### 2.2 Markov Random Fields

An MRF is an undirected graph over variables $\mathbf{X}$, populated with factors. The MRF parameters are given by the vector $\theta = (\ldots, \theta_{\mathbf{X}_f}, \ldots)$, where $\mathbf{X}_f$ are the variables of factor $f$. Component $\theta_{\mathbf{X}_f}$ is a parameter set for a factor $f$, assigning a number $\theta_{\mathbf{x}_f} > 0$ for each instantiation $\mathbf{x}_f$ of variable set $\mathbf{X}_f$.

Given a dataset $\mathcal{D}$ with examples $\mathbf{d}_1, \ldots, \mathbf{d}_N$, the *log likelihood* of parameter estimates $\theta$ is defined as:

$$\ell\ell(\theta|\mathcal{D}) = \sum_{i=1}^{N} \log Z_\theta(\mathbf{d}_i) - N \log Z_\theta. \tag{1}$$

Here, $Z_\theta$ is the partition function, $Z_\theta = \sum_{\mathbf{x}} \prod_f \theta_{\mathbf{x}_f}$ and $Z_\theta(\mathbf{d}_i) = \sum_{\mathbf{x} \sim \mathbf{d}_i} \prod_f \theta_{\mathbf{x}_f}$ ($\mathbf{d}_i \sim \mathbf{x}$ means that instantiations $\mathbf{d}_i$ and $\mathbf{x}$ are compatible). For simplicity, we will assume a tabular representation of factors as opposed to an exponential representation as given in [24, Chapter 19]. In our experiments, however, we use the exponential representation to avoid the need for explicit non-negativity constraints.

The first term in Equation 1 is called the data term, whereas the second term is called the model term. If the data is complete, Equation 1 can be formulated as a convex optimization problem, and the data term becomes trivial to evaluate. However, when the data is incomplete, Equation 1 is non-convex.

### 2.3 Bayesian Networks

A Bayesian network is a directed acyclic graph populated with conditional probability tables (CPTs). Generally, we will use $X$ to denote a variable in a Bayesian network and $\mathbf{U}$ to denote its parents. For every variable instantiation $x$ and parent instantiation $\mathbf{u}$, the Bayesian network includes a parameter $\theta_{x|\mathbf{u}}$ that represents the probability $Pr(X = x|\mathbf{U} = \mathbf{u})$. This implies the requirement that $\sum_x \theta_{x|\mathbf{u}} = 1$, for each parent instantiation $\mathbf{u}$.

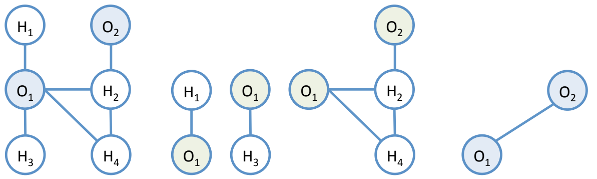
Given a dataset $\mathcal{D}$ with examples $\mathbf{d}_1, \ldots, \mathbf{d}_N$, the *log likelihood* of parameter estimates $\theta$ is defined as:

$$\ell\ell(\theta|\mathcal{D}) = \textstyle\sum_{i=1}^{N} \log Pr_\theta(\mathbf{d}_i).$$

Here, $Pr_\theta$ is the distribution induced by the network structure and parameters $\theta$. One typically seeks maximum likelihood parameters

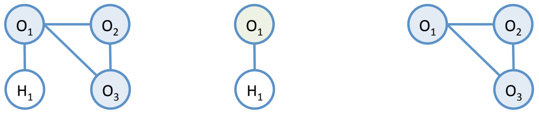
$$\theta^\star = \underset{\theta}{\operatorname{argmax}}\, \ell\ell(\theta|\mathcal{D}).$$

It is not uncommon to also assume a Dirichlet prior on network parameters. In particular, for each variable $X$ with values $x_1, \ldots, x_n$, and parent instantiation $\mathbf{u}$, a Dirichlet prior is specified using exponents $\psi_{x_1|\mathbf{u}}, \ldots, \psi_{x_n|\mathbf{u}}$. This prior induces a density $\propto \prod_{i=1}^{n} [\theta_{x_i|\mathbf{u}}]^{\psi_{x_i|\mathbf{u}} - 1}$ over the parameters $\theta_{x_1|\mathbf{u}}, \ldots, \theta_{x_n|\mathbf{u}}$ of variable $X$ given parent instantiation $\mathbf{u}$. It is also common to assume that exponents are $> 1$, which guarantees a unimodal density. With



(a) Graph (b) Components (c) Auxiliary graph

Figure 1: Auxiliary MRF graph under fully observed variables $\mathbf{O} = \{O_1, O_2\}$.



(a) Graph (b) Components (c) Auxiliary graph

Figure 2: Auxiliary MRF graph under fully observed variables $\mathbf{O} = \{O_1, O_2, O_3\}$.

Dirichlet priors, the objective function becomes

$$\ell\ell(\theta|\mathcal{D}) + \log \rho(\theta).$$

Here, $\rho(\theta)$ is proportional to the prior density on parameters $\theta$, and is given by

$$\rho(\theta) = \prod_{X\mathbf{u}} \prod_{x} [\theta_{x|\mathbf{u}}]^{\psi_{x|\mathbf{u}}-1}.$$

Parameters that optimize the above objective function are called MAP estimates as they maximize the posterior density of the parameters given the dataset.

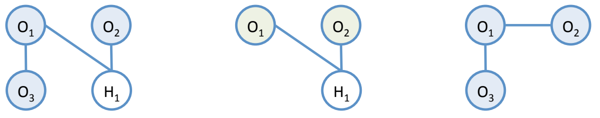
When every exponent $\psi_{x|\mathbf{u}}$ is equal to 1 (uninformative prior), we get $\rho(\theta) = 1$ and MAP estimates reduce to maximum likelihood estimates. Moreover, when every exponent $\psi_{x|\mathbf{u}}$ is equal to 2, MAP estimates reduce to maximum likelihood estimates with Laplace smoothing. This is a common technique to deal with the problem of insufficient counts (i.e., instantiations that never appear in the dataset, leading to zero probabilities and division by zero). We will use Laplace smoothing in our experiments.

## 3 An Upper Bound for MRFs

In this section, we utilize variables that are always observed in a dataset to obtain an upper bound on the likelihood. The bound is obtained by solving a convex optimization problem over an auxiliary MRF, which is defined next.

### 3.1 Decomposition

The auxiliary MRF is obtained by first decomposing the MRF graph $G$ using variables $\mathbf{O}$ that are fully observed in the dataset.



(a) Graph (b) Components (c) Auxiliary graph

Figure 3: Auxiliary MRF graph under fully observed variables $\mathbf{O} = \{O_1, O_2, O_3\}$.

**Definition 1** *Let $G|\mathbf{O}$ be the result of deleting variables $\mathbf{O}$ from graph $G$. A component of $G|\mathbf{O}$ is a maximal set of nodes $\mathbf{S}$ that are connected in $G|\mathbf{O}$. A variable $B$ is a boundary for component $\mathbf{S}$ iff edge $B - S$ appears in $G$, $B \notin \mathbf{S}$ and $S \in \mathbf{S}$.*

Boundary variables must be included in $\mathbf{O}$. Moreover, component variables cannot intersect with $\mathbf{O}$. Figures 1–3 depict the components and boundaries of some MRF graphs.

We are now ready to define the auxiliary MRF by defining its graph. The auxiliary MRF will then have one factor over each maximal clique of this graph.

**Definition 2** *The auxiliary graph for graph $G$ and variables $\mathbf{O}$ is denoted $A_{G|\mathbf{O}}$ and defined as follows: (1) The nodes of $A_{G|\mathbf{O}}$ are the variables $\mathbf{O}$; (2) $A_{G|\mathbf{O}}$ has an edge $X - Y$ iff the edge exists in $G$ or $X$ and $Y$ are boundary variables of some component of $G|\mathbf{O}$.*

Figures 1–3(c) depict some auxiliary MRF graphs.

### 3.2 Optimization

We will next use the auxiliary MRF graph to formulate a convex optimization problem, called the auxiliary problem. The solution of this auxiliary problem will provide an upper bound on the likelihood.

**Definition 3** *Given an MRF graph $G$, and a corresponding dataset $\mathcal{D}$ with fully observed variables $\mathbf{O}$, the auxiliary optimization problem is that of learning the parameters of auxiliary MRF $A_{G|\mathbf{O}}$ from dataset $\mathcal{D}_\mathbf{O}$.*

The auxiliary optimization problem is always convex. This follows since the graph $A_{G|\mathbf{O}}$ contains only variables $\mathbf{O}$, which are fully observed in the dataset $\mathcal{D}$. Hence, the auxiliary optimization problem corresponds to learning the parameters of an MRF under complete data ($\mathcal{D}_\mathbf{O}$ is a complete dataset in this case).

The following theorem shows that the solution of the convex optimization problem provides an upper bound on the likelihood.

**Theorem 1** *Let $G$ be an MRF graph and $\mathcal{D}$ be a corre-*

*sponding dataset with fully observed variables* $\mathbf{O}$*. Let* $f(\theta)$ *be the likelihood function for MRF graph* $G$*, and let* $g(\theta)$ *be the likelihood function for its auxiliary MRF graph* $A_{G|\mathbf{O}}$*. We then have* $f(\theta) \leq g(\theta^*)$*, where* $\theta^*$ *is the global optimum for* $g(\theta)$*.*

**Proof** Let $F_1(\mathbf{X}_1), \ldots, F_n(\mathbf{X}_n)$ be the factors of auxiliary MRF $A_{G|\mathbf{O}}$, representing parameters $\theta$ (i.e., each $F_j(\mathbf{x}_j)$ is a parameter in $\theta$). Note that $\mathbf{X}_j$ is a maximal clique of the auxiliary graph and $\mathbf{X}_j \subseteq \mathbf{O}$. Let the dataset $\mathcal{D}$ be $\{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$. The convex optimization problem $g(\theta)$ is then

$$\text{maximize} \quad g(\theta) = \sum_{i=1}^{N} \log Z_\theta(\mathbf{d}_i) - N \log Z_\theta \qquad (2)$$

where

$$\begin{aligned} Z_\theta &= \sum_{\mathbf{o}} \prod_{j=1}^{n} F_j(\mathbf{x}_j), \quad \mathbf{x}_j \sim \mathbf{o} \qquad (3) \\ Z_\theta(\mathbf{d}_i) &= \prod_{j=1}^{n} F_j(\mathbf{x}_j), \qquad \mathbf{x}_j \sim \mathbf{d}_i \qquad (4) \end{aligned}$$

We will now expand the above equations for optimizing the auxiliary likelihood $g(\theta)$ so we can optimize the original likelihood $f(\theta)$. The basic observation is that $f(\theta)$ can be written in terms of marginals over the fully observed variables. However, these marginals are not free to take any values, as they have to correspond to some original parameters that realize such marginals. Hence, we must constrain these marginals, which correspond to auxiliary parameters $F_i(\mathbf{x}_i)$, in terms of the original parameters. We do this next.

First, note that for each factor $f_k(\mathbf{Y}_k)$ of the original MRF $G$, there is some factor $F_j(\mathbf{X}_j)$ of the auxiliary MRF, such that $\mathbf{Y}_k \cap \mathbf{O} \subseteq \mathbf{X}_j$. We will therefore assign each original factor $f_k$ to a corresponding auxiliary factor $F_j$, writing $f_k^j$ to denote this assignment.

Next, for each auxiliary factor $F_j(\mathbf{X}_j)$, let $\mathbf{Z}_j$ be the variables appearing in original factors $f_k^j$, but not in the auxiliary factor $F_j$. Consider now the following equation, which defines auxiliary parameters $F_j(\mathbf{x}_j)$ in terms of original parameters $f_k^j(\mathbf{y}_k)$:

$$F_j(\mathbf{x}_j) = \sum_{\mathbf{z}_j} \prod_{k} f_k^j(\mathbf{y}_k), \qquad \mathbf{y}_k \sim \mathbf{x}_j \mathbf{z}_j \qquad (5)$$

The original optimization problem $f(\theta)$ can now be defined using Equations 2, 3 and 4, subject to the equality constraints of Equation 5 (i.e., we are now optimizing the original parameters $f_k^j(\mathbf{y}_k)$). By relaxing these equality constraints, and optimizing over the auxiliary parameters $F_j(\mathbf{x}_j)$, we get back the auxiliary optimization problem. Since the latter is obtained by relaxing constraints, we have $f(\theta) \leq g(\theta^\star)$. □

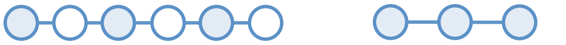


Figure 4: A chain MRF with alternating fully observed variables, and its corresponding auxiliary MRF.

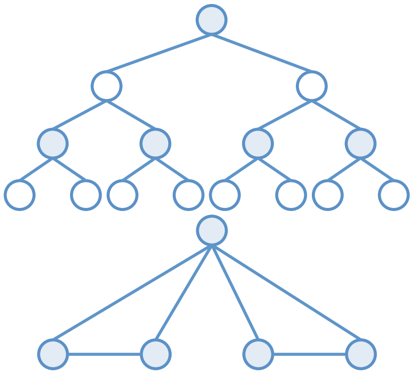


Figure 5: A binary tree MRF with alternating fully observed levels, and its corresponding auxiliary MRF.

Note that when all variables are fully observed in the dataset $\mathcal{D}$ (i.e., the dataset is complete), the auxiliary MRF graph corresponds to the original MRF graph, and the bound becomes exact.

### 3.3 Computing the Bound

The proposed upper bound can be computed using standard methods for estimating parameters under complete data. These methods require inference on the auxiliary MRF, whose complexity depends on the treewidth of its underlying graph. This treewidth can be larger or smaller than the treewidth of the original MRF, depending on the patterns of data incompleteness. We will illustrate this next using a set of examples, in which fully observed nodes are shaded, while hidden nodes are left unshaded.

Figures 4 and 5 show MRFs of bounded treewidth and certain patterns of data incompleteness that lead to auxiliary MRFs with bounded treewidth. In particular, Figure 4 shows a chain with alternating fully observed and hidden variables, which results in an auxiliary MRF with treewidth 1, regardless of the chain length. Figure 5 shows a complete binary tree with alternating fully observed levels, leading to an auxiliary MRF with treewidth 2, regardless of the tree depth.

Figure 6 shows an example where the auxiliary MRF has a lower treewidth. However, Figure 7 shows an $n \times n$ grid, leading to an auxiliary MRF with treewidth $2n-1$. Similarly, Figure 8 shows an MRF with treewidth 1, yet an auxiliary MRF of treewidth $n$.

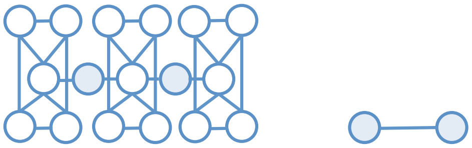


Figure 6: An MRF structure that leads to an auxiliary MRF of lower treewidth.

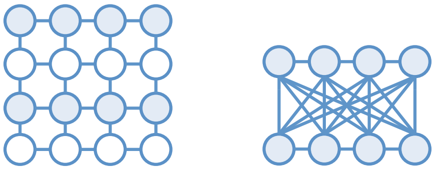


Figure 7: A grid with alternating fully observed rows, and its corresponding auxiliary MRF.

## 4 An Upper Bound for Bayesian Networks

We now present a similar upper bound on the likelihood function for a Bayesian network structure $G$. Again, the bound is defined based on the set of fully observed variables $\mathbf{O}$ in a dataset.

**Definition 4 ([26])** *Let $G|\mathbf{O}$ be the result of deleting edges in DAG $G$ that are outgoing from variables $\mathbf{O}$. A component of $G|\mathbf{O}$ is a maximal set of variables $\mathbf{S}$ that are connected in $G|\mathbf{O}$. A variable $B$ is a boundary for component $\mathbf{S}$ iff edge $B \rightarrow S$ appears in $G$, $B \notin \mathbf{S}$ and $S \in \mathbf{S}$.*

Figure 9 depicts an example DAG with its components and boundaries. Note that the boundary variables $\mathbf{B}$ of a component must all be fully observed, $\mathbf{B} \subseteq \mathbf{O}$. Moreover, for any component $\mathbf{S}$, the variables $\mathbf{S} \cap \mathbf{O}$ must be leaf nodes in $G|\mathbf{O}$.

We will next interpret the boundary variables $\mathbf{B}$ of each component $\mathbf{S}$ as the parents of observed variables in component $\mathbf{S}$. This interpretation will be used to define an auxiliary distribution over the fully observed variables.

**Definition 5** *The auxiliary distribution for DAG $G$ and variables $\mathbf{O}$ is denoted $P_{G|\mathbf{O}}$ and defined as follows: (1) $P_{G|\mathbf{O}}$ is over the variables $\mathbf{O}$, (2) $P_{G|\mathbf{O}}$ is the product of factors $Pr(\mathbf{L}|\mathbf{B})$, where $\mathbf{B}$ is the boundary variables of some component $\mathbf{S}$ and $\mathbf{L} = \mathbf{S} \cap \mathbf{O}$.*

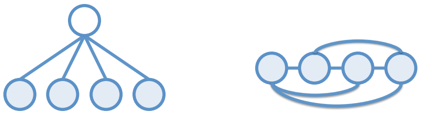
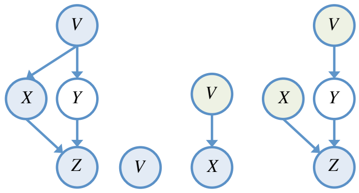


Figure 8: An MRF structure with treewidth 1, and its corresponding auxiliary MRF of treewidth $n$.



(a) DAG (b) Components

Figure 9: A DAG and its components under fully observed variables $\mathbf{O} = \{V, X, Z\}$.

Each probability $Pr(\mathbf{l}|\mathbf{b})$ will be interpreted as an auxiliary parameter $\theta_{\mathbf{l}|\mathbf{b}}$. We can now state our upper bound.

**Theorem 2** *Let $G$ be a DAG and $\mathcal{D}$ be a corresponding dataset with fully observed variables $\mathbf{O}$. Let $f(\theta)$ be the likelihood function for $G$ and $\mathcal{D}$, and let $g(\theta)$ be the likelihood function for the auxiliary distribution $P_{G|\mathbf{O}}$. We then have $f(\theta) \leq g(\theta^*)$, where $\theta^*$ is the global optimum for $g(\theta)$.*

**Proof** We now sketch the proof of this theorem, which is similar to the one for MRFs. That is, we express auxiliary parameters in terms of original parameters, allowing us to formulate the original optimization problem as an optimization problem with non-convex equality constraints (which relate auxiliary and original parameters). By relaxing these equality constraints, we obtain a convex optimization problem that provides an upper bound on the original optimization problem. Hence, it suffices to show the non-convex equality constraints in this case.

Consider a factor $Pr(\mathbf{X}\ |\mathbf{U})$ of the auxiliary distribution, and the corresponding parameters $\theta_{\mathbf{x}|\mathbf{u}}$. Variables $\mathbf{X}$ must then be leaves of some component $\mathbf{S}$ in $G|\mathbf{O}$, and $\mathbf{U}$ must correspond to the boundary variables of component $\mathbf{S}$. One can then express each auxiliary parameter $\theta_{\mathbf{x}|\mathbf{u}}$ in terms of original parameters that pertain only to the variables in component $\mathbf{S}$. In particular, let $Pr(.)$ be the distribution induced by the original DAG $G$ and let $\mathbf{y}$ be an instantiation of variables $\mathbf{S} \setminus \mathbf{X}$. We then have $Pr(\mathbf{x}|\mathbf{u}) = \sum_{\mathbf{y}} Pr(\mathbf{x}, \mathbf{y}|\mathbf{u})$. Moreover, $Pr(\mathbf{x}, \mathbf{y}|\mathbf{u})$ can be expressed in terms of original parameters pertaining only to variables $\mathbf{S}$. This follows since, given $\mathbf{U}$, $\mathbf{S}$ is independent of all other variables in DAG $G$. □

One difference from the upper bound for MRFs is that this bound can be computed more efficiently. In particular, the optimal estimate $\theta^\star$ can be identified using a single pass through the dataset $\mathcal{D}_{\mathbf{O}}$. Similarly, $g(\theta^\star)$ can be computed using a single pass through the dataset, once the estimate $\theta^\star$ is identified.

## 5 Experimental Results

Our experiments are structured as follows. Given a network $G$, we generate a dataset $\mathcal{D}$ while ensuring that a certain percentage of variables are fully observed, with all others hidden. Using dataset $\mathcal{D}$, we estimate the parameters of network $G$ using EM.

We compare the local optimum learned by EM, to the proposed bound gotten using decomposition, and to the bound that assumes all distributions are valid (which we call the naive bound).

The naive bound is computed by discarding the graph structure and assigning a probability to every data example based on its number of occurrences in the dataset. This effectively assumes a fully connected graph. Consider, for example, a simple dataset with a data example $\mathbf{d}_1$ that is repeated twice and another $\mathbf{d}_2$ that is repeated 3 times. The naive bound assigns a probability $\frac{2}{5}$ to $\mathbf{d}_1$, and $\frac{3}{5}$ to $\mathbf{d}_2$, and computes the likelihood: $(\frac{2}{5})^2 \times (\frac{3}{5})^3$.

Before we present our results, we have the following observations on our data generation model. First, we made all unobserved variables hidden (as opposed to missing at random) as this leads to a more difficult learning problem, especially for EM. Second, it is not uncommon to have a significant number of variables that are always observed in real-world datasets. For example, in the UCI repository: the internet advertisements dataset has 1558 variables, only 3 of which have missing values; the automobile dataset has 26 variables, where 7 have missing values; the dermatology dataset has 34 variables, where only age can be missing; and the mushroom dataset has 22 variables, where only one variable has missing values [1].

In our experiments, we use the following networks: alarm, andes, asia, win95pts, diagnose, pigs, spect, water, together with chains, trees, and grids. Network win95pts (76 variables) is an expert system for printer troubleshooting in Windows 95, whereas Network pigs is used for diagnosing the PSE disease. Network andes is an intelligent tutoring system. Network diagnose is from the UAI 2008 evaluation. Network spect is a naive Bayes network induced from a dataset in the UCI ML repository, with 1 class variable and 22 attributes. Chains, trees, and grids are randomly generated networks. The other networks are commonly used benchmarks.

Figures from 10 to  25 show the objective function values gotten by EM for different benchmarks and for different percentages of fully observed variables, together with the proposed bound (decomposition bound), and the naive bound. We have the following observations on the results.

In most cases, our proposed bound was much tighter than the naive bound. One can also see that the proposed bound coincides (or almost coincides) with the EM's curve in Figures 11, 15, 18, 19, 20, 21, 22, 23, 24, and 25. This shows that the bound can be tight in many cases. When the bound coincides with the EM curve, it provides a certificate that EM is getting the global optimum in these cases.

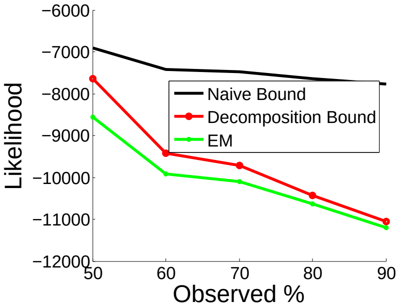


Figure 10: Upper bound for network Alarm.

Moreover, in Figures 10, 14, and 17, as the number of fully observed variables increases, the gap between the proposed bound and EM's curve tends to shrink, which suggests that the bound becomes tight and that EM gets close to the global optimum in these cases. On the other hand, for cases where the proposed bound was not close to the EM's curve, it could be that EM is getting a local optimum, or the bound is not tight, in these cases.

Furthermore, we note that the excellent performance of the upper bound on Network Spect in Figure 15, and on tree networks in Figures 20, 21, and 22 is partially because hidden variables associated with leaf nodes in these networks can be ignored from the computation of the likelihood, as their values are summed out.

We finally conduct an experiment to see how often EM approaches the bound if started from different seeds; using a $3 \times 3$ grid while fully observing $50\%$ of the variables. Figure 26 shows the difference in likelihood between the upper bound and EM for this benchmark, when started from different seeds (x-axis). One can see that EM gets close to the bound in many cases for this benchmark. We note, however, that a more comprehensive study is needed for assessing the quality of EM estimates under different seeds—a study that can be significantly aided by the proposed upper bound.

## 6 Related Work

Decomposing Bayesian networks based on fully observed variables was proposed in [26] to speed-up parameter estimation. Our bound relies on this decomposition as a first step in formulating the auxiliary optimization problem.

Variational methods (see [14]) can provide lower bounds on the likelihood in graphical models. Moreover, an upper bound for the likelihood in the context of Gaussian mixtures was proposed in [3]. However, this bound only works

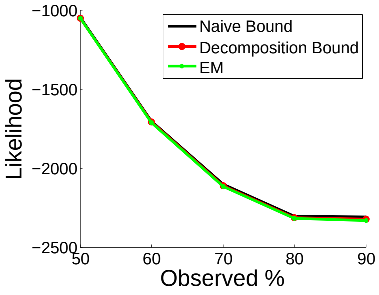


Figure 11: Upper bound for network Asia.

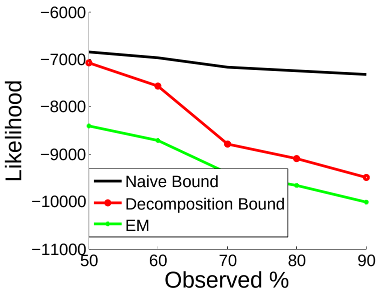


Figure 12: Upper bound for network Win95pts.

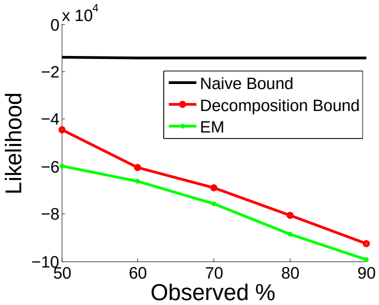


Figure 13: Upper bound for network Diagnose.

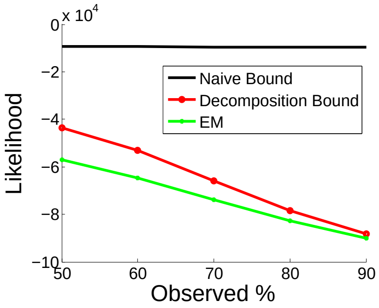


Figure 14: Upper bound for network Andes.

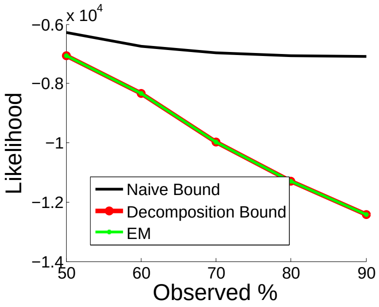


Figure 15: Upper bound for network Spect.

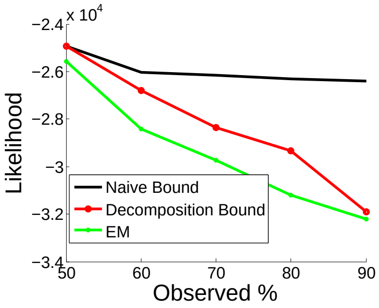


Figure 16: Upper bound for network Water.

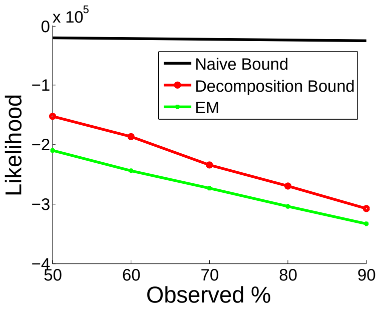


Figure 17: Upper bound for network Pigs.

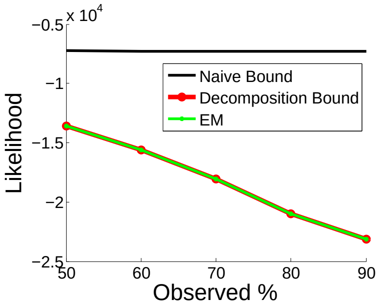


Figure 18: Upper bound for a chain network (50 nodes).

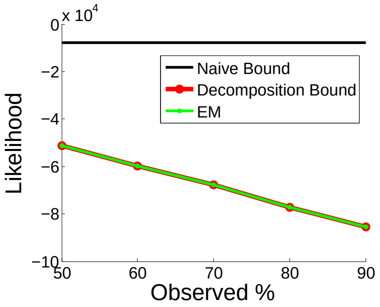


Figure 19: Upper bound for a chain network (180 nodes).

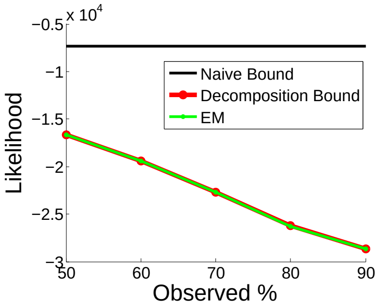


Figure 20: Upper bound for a tree network (63 nodes).

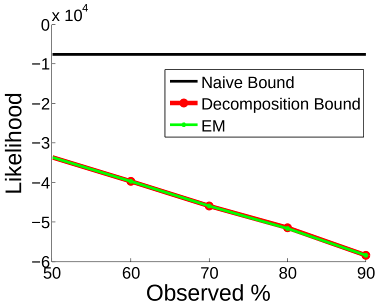


Figure 21: Upper bound for a tree network (127 nodes).

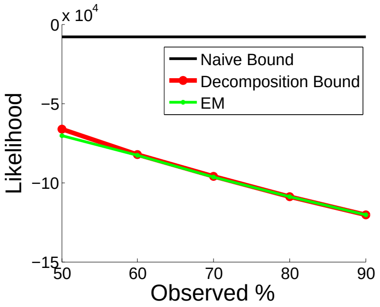


Figure 22: Upper bound for a tree network (255 nodes).

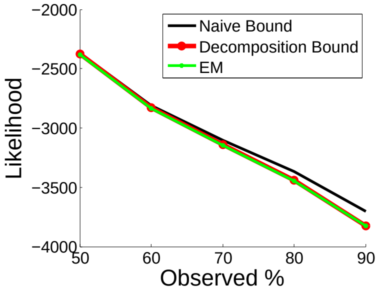


Figure 23: Upper bound for a $3 \times 3$ MRF grid.

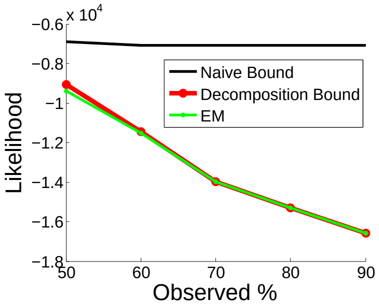


Figure 24: Upper bound for a $6 \times 6$ MRF grid.

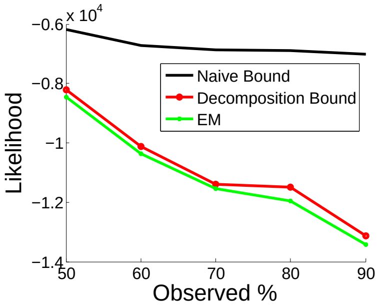


Figure 25: Upper bound for a $9 \times 9$ MRF grid.

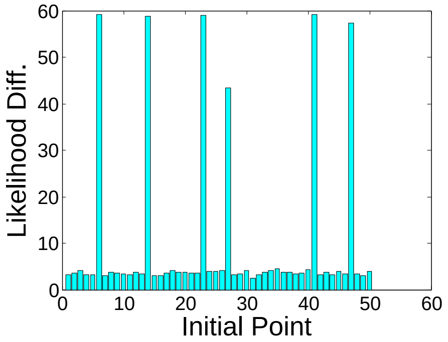


Figure 26: Likelihood difference between decomposition bound and EM started from different points.

asymptotically. An upper bound on maximum likelihood that only works for phylogenetic trees was proposed in [7]. Techniques for computing upper and lower bounds on likelihoods in sigmoid and noisy-OR networks were proposed in [12].

Some work also exists for obtaining upper and lower bounds on the partition function. In particular, mean field theory, e.g. [33, 14], provides such a lower bound (tighter bounds have also been derived [19]). In contrast, upper bounds are not widely available [31]. For the special case of the Ising Model, a recursive procedure was proposed for upper bounding the log partition function [11]. An upper bound on the partition function of an arbitrary MRF was proposed in [31] based on solving a convex variational problem. While bounds on the partition function can be used to get an upper bound on the likelihood, the non-convex term related to the data remains non-convex, which does not make the bound easy to compute. The bound we proposed, however, is based on solving a convex optimization problem.

## 7 Conclusion

We proposed a technique for obtaining an upper bound on the global optimum in parameter estimation. The technique applies to incomplete datasets and exploits variables that are always observed in the dataset. The bound is computed by solving a convex optimization problem, which can be solved by a single pass through the dataset in Bayesian networks. The proposed bound can be useful in providing a certificate of global optimality for parameters learned by estimation algorithms. Empirically, we showed that the bound can be tight, and can be used to show that an estimation algorithm is obtaining the global optimum or an estimate that is very close to the optimum.

## Acknowledgments

This work was supported by ONR grant #N00014-12-1-0423 and NSF grant #IIS-1118122.

## References

[1] K. Bache and M. Lichman. Uci machine learning repository. Technical report, Irvine, CA: University of California, School of Information and Computer Science, 2013.

[2] J. Besag. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24:179–195, 1975.

[3] Christophe Biernacki. An asymptotic upper bound of the likelihood to prevent gaussian mixtures from degenerating. Technical report, Université de Franche-Comté, 2004.

[4] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[6] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London Series*, 1922.

[7] Michael D. Hendy and Barbara R. Holland. Upper bounds on maximum likelihood for phylogenetic trees. *Bioinformatics*, 2003.

[8] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Research of the National Bureau of Standards*, 1952.

[9] G. Hinton. Training products of experts by minimizing contrastive divergence. In *Neural Computation*, 2000.

[10] A. Hyvärinen. Estimation of non-normalized statistical models using score matching. *JMLR*, 2005.

[11] T. S. Jaakkola and M. Jordan. Recursive algorithms for approximating probabilities in graphical models. In *Advances in Neural Information Processing Systems*, 1996.

[12] Tommi S. Jaakkola and Michael I. Jordan. Computing upper and lower bounds on likelihoods in intractable networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1998.

[13] Radim Jirousek and Stanislav Preucil. On the effective implementation of the iterative proportional fitting procedure. *Computational Statistics & Data Analysis*, 19(2):177–189, 1995.

[14] M. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. Saul. *Learning in Graphical Models*, chapter "An introduction to variational methods for graphical models. Cambridge, MA: MIT Press, 1999.

[15] R. Kindermann and J. L. Snell. *Markov Random Fields and their Applications*. American Mathematical Society, 1980.

[16] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[17] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[18] S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.

[19] S. L. Lauritzen. *Graphical Models*. Oxford, U.K.: Oxford Univ. Press, 1996.

[20] S Z. Li. Markov random field modeling in image analysis. *Springer-Verlag*, 2001.

[21] D. C. Liu and J. Nocedal. On the Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming*, 45(3):503–528, 1989.

[22] E. Marinari, G. Parisi, and J.J. Ruiz-Lorenzo. Numerical simulations of spin glass systems. *Spin Glasses and Random Fields*, 1997.

[23] Yariv Dror Mizrahi, Misha Denil, and Nando de Freitas. Linear and parallel learning of Markov random fields. In *In International Conference on Machine Learning (ICML)*, 2014.

[24] Kevin Patrick Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[25] Khaled S. Refaat, Arthur Choi, and Adnan Darwiche. EDML for learning parameters in directed and undirected graphical models. In *Advances in Neural Information Processing Systems 26*, pages 1502–1510, 2013.

[26] Khaled S. Refaat, Arthur Choi, and Adnan Darwiche. Decomposing parameter estimation problems. In *Advances in Neural Information Processing Systems 27*, pages 1565–1573, 2014.

[27] Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 1996.

[28] S. Russel, J. Binder, D. Koller, and K. Kanazawa. Local learning in probabilistic networks with hidden variables. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[29] R. Shachter. Evidence absorption and propagation through evidence reversals. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1990.

[30] C. Varin, N. Reid, and D Firth. An overview of composite likelihood methods. *Statistica Sinica*, 2011.

[31] Martin J. Wainwright, Tommi S. Jaakkola, and IEEE Alan S. Willsky, Fellow. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 2005.

[32] C. Yanover, O. Schueler-Furman, and Y. Weiss. Minimizing and learning energy functions for side-chain prediction. *In Speed, Terry and Huang, Haiyan (eds.), Research in Computational Molecular Biology, volume 4453 of Lecture Notes in Computer Science*, 2007.

[33] J. Zhang. The application of the gibbs-bogoliubov-feynman inequality in mean-field calculations for Markov random-fields. *IEEE Tran. on Image Process.*, 5(7):1208–1214, 1996.