An Efficient Method for Bayesian Network Parameter Learning from Incomplete Data

Karthika Mohan KARTHIKA@CS.UCLA.EDU
Guy Van den Broeck GUYVDB@CS.UCLA.EDU
Arthur Choi AYCHOI@CS.UCLA.EDU
Judea Pearl JUDEA@CS.UCLA.EDU
Computer Science Department, University of California, Los Angeles

Abstract

We propose an efficient method for estimating the parameters of a Bayesian network, from incomplete datasets, i.e., datasets containing variables with missing values. In contrast to textbook approaches such as EM and the gradient method, our approach is non-iterative, yields closed form parameter estimates, and eliminates the need for inference in a Bayesian network. Our approach is capable of producing consistent parameter estimates for missing data problems that are MCAR, MAR, and in some cases, MNAR. Empirically, our approach is orders of magnitude faster than EM. When data is scarce, we learn parameters of comparable quality to EM. Given sufficient data, we can learn parameters that are orders of magnitude closer to the true parameters.

1. Introduction

When learning the parameters of a Bayesian network from data with missing values, the conventional wisdom among machine learning practitioners is that there are two options: either use *expectation maximization* (EM) or use likelihood *optimization* with a gradient method; see, e.g., (Darwiche, 2009; Koller & Friedman, 2009; Murphy, 2012; Barber, 2012). These two approaches are known to consistently estimate the parameters when values in the data are *missing at random* (MAR). However, these two standard approaches suffer from the following disadvantages. First, they are *iterative*, and hence they may require many passes over a potentially large dataset. Next, they *require inference* in the Bayesian network, which is by itself already intractable (for networks with high treewidth, and not enough local structure (Chavira & Darwiche, 2006; 2007)). Finally, these algorithm may get stuck in *local optima*, which means that, in practice, one must run these algorithms multiple times with different initial seeds (then keep those parameter estimates that obtained the best likelihood).

Recently, Mohan et al. (2013) showed that the joint distribution of a Bayesian network can be recovered consistently from incomplete data, for all MCAR and MAR problems as well as a major subset of MNAR problems, when given access to a formal representation, called a missingness graph, that represents the *causal mechanisms* responsible for missingness in an incomplete dataset. Using this representation, they are able to decide whether there exists a consistent estimator for a given query Q (which can be, for example, a joint or conditional distribution). If the answer is affirmative, they identify *closed form* expressions to estimate Q in terms of the observed data. The estimand obtained enables one to process the observed data directly in a single pass with a guarantee that, as the number of samples increases, the estimate would converge to the true value of Q *as if* no data were missing.

Based on this framework, we contribute a new and *practical family of parameter learning algorithms* for Bayesian networks. Here, we focus on the traditional MCAR and MAR assumptions. The key insight of our work is the following. There exists a most-general, least-committed missingness graph that captures the MCAR or MAR assumption, but invokes no additional independencies. Although this is a minor technical observation, it has far-reaching consequences. It enables the techniques of Mohan et al. to be applied directly to MCAR or MAR data, without requiring the user to provide a more specific missingness graph. Hence, it enables our new algorithms to serve as drop-in replacements for the already influential EM algorithm in existing applications. It results in practical algorithms for learning the parameters of a Bayesian network from an incomplete dataset, which have the following advantages:

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

1. the parameter estimates are *consistent* when the values of a dataset are MCAR or MAR, i.e., we recover the true parameters as the dataset size approaches infinity,
2. the parameter estimates are efficiently computable in *closed-form*, requiring only a single pass over the data,
3. the parameter estimates require *no inference* in the Bayesian network.

Whereas advantage (1) is the same guarantee that EM provides, advantages (2) and (3) are significant computational advantages over EM, in particular when the dataset size is very large (cf., the Big Data paradigm), or for Bayesian networks that are intractable for exact inference. Moreover, because of advantage (2), we do not use iterative optimization, and our estimates do not suffer from local optima. Note further that all these advantages are already available to us when learning Bayesian networks from *complete* datasets, properties which certainly contributed to the popularity of Bayesian networks today, as probabilistic models.

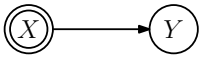
As a secondary contribution, we show how to further factorize estimates to extract more information from the data and thus improve the convergence of our algorithms.

2. Technical Preliminaries

In this paper, we use upper case letters (X) to denote variables and lower case letters (x) to denote their values. Variable sets are denoted by bold-face upper case letters ($\mathbf{X}$) and their instantiations by bold-face lower case letters ($\mathbf{x}$). Generally, we will use X to denote a variable in a Bayesian network and $\mathbf{U}$ to denote its parents. A network parameter will therefore have the general form $\theta_{x|\mathbf{u}}$, representing the probability $Pr(X=x|\mathbf{U}=\mathbf{u})$.

Consider the following dataset $\mathcal{D}$, and the directed acyclic graph (DAG) $\mathcal{G}$ of a Bayesian network, both over two variables, X and Y.

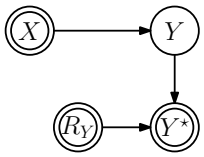
X	Y
$\bar{x}$	y
x	?
x	$\bar{y}$
$\bar{x}$	?
...	



In this example, the value for variable X is always observed in the data, while the value for variable Y can be missing. In the graph, we shall denote a variable that is always observed with a double-circle.

Now, if we know the mechanism that causes the value of Y to become missing in the data, we can include it in our model. For example, consider the following expanded dataset and graph.

X	Y	$Y^\star$	R_Y
$\bar{x}$	y	y	ob
x	?	mi	unob
x	$\bar{y}$	$\bar{y}$	ob
$\bar{x}$	?	mi	unob
...			



Here, we have augmented the dataset and graph with new variables. Variable R_Y represents the causal mechanism that dictates the missingness of the value of Y. This mechanism can be active (Y is unobserved), which we denote by $R_Y = \mathsf{unob}$. Otherwise, the mechanism is passive (Y is observed), which we denote by $R_Y = \mathsf{ob}$. Variable $Y^\star$ acts as a proxy on the value of Y in the data, which may be an observed value y, or a special value (mi), value when the value of Y is missing. The value of $Y^\star$ thus depends functionally on the values of R_Y and Y:

$$Y^\star = f(R_Y, Y) = \begin{cases} \mathsf{mi} & \text{if } R_Y = \mathsf{unob} \\ Y & \text{if } R_Y = \mathsf{ob} \end{cases}$$

That is, when $R_Y = \mathsf{unob}$, then $Y^\star = \mathsf{mi}$; otherwise $R_Y = \mathsf{ob}$ and the proxy $Y^\star$ assumes the observed value of variable Y. We can encode this also as a CPT:

$$\Pr(Y^\star|Y, R_Y) = \begin{cases} 1 & \text{if } R_Y = \mathsf{unob} \text{ and } Y^\star = \mathsf{mi} \\ 1 & \text{if } R_Y = \mathsf{ob} \text{ and } Y^\star = Y \\ 0 & \text{otherwise.} \end{cases}$$

In general, when we want to learn a Bayesian network $\mathcal{N}$ from an incomplete dataset $\mathcal{D}$, there is an underlying but unknown distribution $\Pr(\mathbf{X})$ that is induced by the network $\mathcal{N}$ that we want to learn. The variables $\mathbf{X}$ are further partitioned into two sets of variables: a set $\mathbf{X}_o$ of fully-observed variables, and a set $\mathbf{X}_m$ of partially-observed variables that have missing values in the data. We can take into account the presence of the causal mechanisms that cause the values of variables $\mathbf{X}_m$ to go missing, as in our example above, by introducing variables $\mathbf{R}$ representing the causal mechanisms themselves, and variables $\mathbf{X}^\star_m$ that act as proxies to the variables $\mathbf{X}_m$. This augmented Bayesian network, which we refer to as $\mathcal{N}^\star$, now has variables $\mathbf{X}_o, \mathbf{X}^\star_m, \mathbf{R}$ that are fully-observed, and variables $\mathbf{X}_m$ that are only partially-observed. Moreover, network $\mathcal{N}^\star$ induces a distribution $\Pr(\mathbf{X}_o, \mathbf{X}_m, \mathbf{X}^\star_m, \mathbf{R})$ which now embeds the original distribution $\Pr(\mathbf{X}_o, \mathbf{X}_m)$ of network $\mathcal{N}$ as a marginal distribution.

Recently, Mohan et al. (2013) identified conditions on the augmented network $\mathcal{N}^\star$ (i.e., the missingness graph) that allow the original, partially-observed distribution $\Pr(\mathbf{X}_o, \mathbf{X}_m)$ to be identified from the fully-observed distribution $\Pr(\mathbf{X}_o, \mathbf{X}^\star_m, \mathbf{R})$. In practice, however, we have access only to an incomplete dataset $\mathcal{D}$, and the correspond-

ing *data distribution* that it induces:

$$\Pr_{\mathcal{D}}(\mathbf{x}_o, \mathbf{x}^\star_m, \mathbf{r}) = \frac{1}{N}\mathcal{D}\#(\mathbf{x}_o, \mathbf{x}^\star_m, \mathbf{r})$$

where N is the number of instances in dataset $\mathcal{D}$, and where $\mathcal{D}\#(\mathbf{x})$ is the number of instances where instantiation $\mathbf{x}$ appears in the data.[1] Moreover, the data distribution $\Pr_{\mathcal{D}}$ tends to the true distribution $\Pr$ (over the fully-observed variables), as the data set size N tends to ∞.

In the following few sections, we show how we can leverage the results of Mohan et al. (2013), identifying in particular some practical and efficient algorithms for the consistent estimation of a Bayesian network's parameters. First, we do not assume a particular missingness graph (which specifies the direct causes, or parents, of the missingness mechanisms $\mathbf{R}$), but instead assume only some general conditions on them that lead to broad classes of missingness graphs, which further characterize commonly-used assumptions on missing data. Subsequently, we will show how to exploit knowledge of the underlying missingness graph that is available (say, from a domain expert), to obtain improved parameter estimates.

2.1. Missingness Categories

An incomplete dataset is categorized as *Missing Completely At Random* (MCAR) if all mechanisms $\mathbf{R}$, that cause the values of variables $\mathbf{X}_m$ to go missing, are marginally independent of $\mathbf{X}$, i.e., $(\mathbf{X}_m, \mathbf{X}_o) \perp\!\!\!\perp \mathbf{R}$. This corresponds to a missingness graph where no variable in $\mathbf{X}_m \cup \mathbf{X}_o$ is a parent of any variable in $\mathbf{R}$. Note that the example graph in Section 2 implies an MCAR dataset.

An incomplete dataset is categorized as *Missing At Random* (MAR) if missingness mechanisms are conditionally independent of the partially-observed variables given the fully-observed variables, i.e., if $\mathbf{X}_m \perp\!\!\!\perp \mathbf{R} \mid \mathbf{X}_o$. This corresponds to a missingness graph where variables $\mathbf{R}$ are allowed to have parents, as long as none of them are partially-observed. In the example missingness graph of Section 2, adding an edge $X \rightarrow R_Y$ results in a graph that yields MAR data. This is a stronger, variable-level definition of MAR, which has previously been used in the machine learning literature (Darwiche, 2009; Koller & Friedman, 2009), in contrast to the event-level definition of MAR, that is prevalent in the statistics literature (Rubin, 1976).

An incomplete dataset is categorized as *Missing Not At Random* (MNAR) if it is neither MCAR nor MAR. In the example missingness graph of Section 2, adding an edge $Y \rightarrow R_Y$ results in a graph that generates MNAR data.

[1] Note that the data distribution is well-defined over the variables $\mathbf{X}_o$, $\mathbf{X}^\star_m$ and $\mathbf{R}$, as they are fully-observed in the augmented dataset. Further, the distribution $\Pr_{\mathcal{D}}$ can be represented compactly in space that is linear in N, as we need not explicitly represent those instantiations $\mathbf{x}$ that were not observed in the data.

3. Closed-Form Learning Algorithms

We will now present a set of algorithms to learn the parameters of a Bayesian network $\mathcal{N}$ from a data distribution $\Pr_{\mathcal{D}}$ (over the fully-observed variables in the augmented dataset). We do so for different missing data assumptions, but without knowing the missingness graph that generated the data. To estimate the conditional probabilities $\theta_{x|\mathbf{u}}$ that parameterize a Bayesian network, we estimate the joint distributions $\Pr(X, \mathbf{U})$, which are subsequently normalized. Hence, it suffices, for our discussion, to estimate marginal distributions $\Pr(\mathbf{Y})$, for families $\mathbf{Y} = \{X\} \cup \mathbf{U}$. Here, we let $\mathbf{Y}_o = \mathbf{Y} \cap \mathbf{X}_o$ denote the observed variables of a family $\mathbf{Y}$, and $\mathbf{Y}_m = \mathbf{Y} \cap \mathbf{X}_m$ to denote the partially-observed variables. Further, we let $\mathbf{R}_{\mathbf{Z}} \subseteq \mathbf{R}$ denote the missingness mechanisms for a set of partially-observed variables $\mathbf{Z}$.

3.1. Direct Deletion: MCAR

We begin with the simplest setting and assume that the incomplete dataset is MCAR. In this setting, we can estimate the marginals $\Pr(\mathbf{Y})$ from the data distribution $\Pr_{\mathcal{D}}$ by

$$\begin{aligned}
&\Pr(\mathbf{Y}) \\
&= \Pr(\mathbf{Y}_o, \mathbf{Y}_m | \mathbf{R}_{\mathbf{Y}_m} = \mathsf{ob}) && \text{by } \mathbf{X}_\mathrm{o}, \mathbf{X}_\mathrm{m} \perp\!\!\!\perp \mathbf{R} \\
&= \Pr(\mathbf{Y}_o, \mathbf{Y}^\star_m | \mathbf{R}_{\mathbf{Y}_m} = \mathsf{ob}) && \text{by } \mathbf{X}_\mathrm{m} = \mathbf{X}^\star_\mathrm{m} \text{ when } \mathbf{R} = \mathsf{ob} \\
&\approx \Pr_{\mathcal{D}}(\mathbf{Y}_o, \mathbf{Y}^\star_m | \mathbf{R}_{\mathbf{Y}_m} = \mathsf{ob})
\end{aligned}$$

This suggests that we can simply use the subset of the data where every variable in $\mathbf{Y}$ is observed. Because the data distribution $\Pr_{\mathcal{D}}$ tends to the true distribution $\Pr$ (over the fully-observed variables) as the dataset size tends to ∞, this implies a consistent estimate for the marginals $\Pr(\mathbf{Y})$.

The statistical technique of *listwise deletion* refers to the process of deleting all samples containing missing values. Estimating the joint probability distribution by direct deletion corresponds to listwise deletion (Mohan et al., 2013). However, estimating $\Pr(\mathbf{Y})$ for any $\mathbf{Y} \subsetneq \mathbf{X}$ by listwise deletion yields the estimand $\Pr_{\mathcal{D}}(\mathbf{Y}_o, \mathbf{Y}^\star_m | \mathbf{R}_{\mathbf{X}_m} = \mathsf{ob})$. This estimand is different from the one obtained by direct deletion, which improves on listwise deletion by incorporating more instances in the dataset. When $\mathbf{Y}_m$ contains exactly two variables, direct deletion for MCAR is known in statistics as pairwise deletion or available-case analysis.

3.2. Direct Deletion: MAR

In this section, we introduce the first deletion algorithm for MAR datasets and show how to estimate family marginals $P(\mathbf{Y})$ under the MAR assumption. Let $\mathbf{X}'_o = \mathbf{X}_o \setminus \mathbf{Y}_o$ denote the fully-observed variables outside of the family

variables $\mathbf{Y}$ (and thus $\mathbf{X}_o = \mathbf{Y}_o \cup \mathbf{X}'_o$). We then have

$$\begin{aligned}\Pr(\mathbf{Y}) &= \sum_{\mathbf{X}'_o} \Pr(\mathbf{Y}_o, \mathbf{Y}_m, \mathbf{X}'_o) \\ &= \sum_{\mathbf{X}'_o} \Pr(\mathbf{Y}_m|\mathbf{Y}_o, \mathbf{X}'_o) \Pr(\mathbf{Y}_o, \mathbf{X}'_o) \\ &= \sum_{\mathbf{X}'_o} \Pr(\mathbf{Y}^\star_m|\mathbf{Y}_o, \mathbf{X}'_o, \mathbf{R}_{\mathbf{Y}_m} = \mathsf{ob}) \Pr(\mathbf{Y}_o, \mathbf{X}'_o)\end{aligned}$$

Hence, we can use direct deletion to estimate $\Pr(\mathbf{Y}_m|\mathbf{Y}_o, \mathbf{X}'_o)$. Otherwise, $\Pr(\mathbf{Y}_o, \mathbf{X}'_o)$ is straightforward to estimate as variables $\mathbf{X}_o$ are fully-observed in the data. This leads to the estimates:

$$\Pr(\mathbf{Y}) \approx \sum_{\mathbf{X}'_o} \Pr_{\mathfrak{D}}(\mathbf{Y}^\star_m|\mathbf{Y}_o, \mathbf{X}'_o, \mathbf{R}_{\mathbf{Y}_m} = \mathsf{ob}) \Pr_{\mathfrak{D}}(\mathbf{Y}_o, \mathbf{X}'_o)$$

Again, the data distribution $\Pr_{\mathfrak{D}}$ tends to the true distribution $\Pr$ (over the fully-observed variables) as the dataset size tends to ∞, which implies a consistent estimate for the marginals $\Pr(\mathbf{Y})$, under the MAR assumption.

For the purposes of our paper, we will refer to the simple technique above as *direct deletion*. This will facilitate comparisons other algorithms discussed in the paper.

3.3. Factored Deletion: MCAR

Direct deletion exploits more data than listwise deletion and now we shall discuss factored deletion algorithm that estimates parameters by exploiting more data than direct deletion algorithm.

Let $Y_i, Y_2 ..., Y_{|\mathbf{Y}|}$ be any ordering of variables in $\mathbf{Y}$ and let Y_i denote the i^{th} variable in the ordering. Notice that every ordering yields a unique factorization $\prod_i \Pr(Y_i|Y_{i+1}, ...Y_{|\mathbf{Y}|})$, of the marginal $P(\mathbf{Y})$. Therefore, for a given ordering of variables we can estimate the marginals of the form $\Pr(\mathbf{Y})$ from the data distribution $\Pr_{\mathfrak{D}}$ as shown below:

$$\begin{aligned}&\Pr(\mathbf{Y}) \\ &= \prod_i \Pr(Y_i|Z) \text{ where } Z = \bigcup_{j=i+1}^{|\mathbf{Y}|} \mathbf{Y}_j \\ &= \prod_i \Pr(Y_i|Z, R_{Z'} = \mathsf{ob}) \text{ where } Z' = \mathbf{Y}_m \cap (Z \cup \{Y_i\}) \\ &= \prod_i \Pr(Y'_i|Z^\star_m, Z_o, R_{Z'} = \mathsf{ob}) \\ &\quad \text{where } Y'_i = Y^\star_i \text{ if } Y_i \in \mathbf{X}_m \text{ and } Y'_i = Y_i \text{ otherwise.} \\ &\approx \prod_i \Pr_{\mathfrak{D}}(Y'_i|Z^\star_m, Z_o, R_{Z'} = \mathsf{ob})\end{aligned}$$

Here, each factor is estimated independently on its own subset of the data. When $i = 1$, we use the same subset of data as in direct deletion. However for factors in which $i > 1$, we can potentially use more data for estimation. For example to estimate the factor corresponding to $i = 2$, we can utilize instances in which variable Y_1 is not observed. Moreover, we observe that different orderings yield different factorizations of $\Pr(\mathbf{Y})$ that can potentially use different subsets of the data.

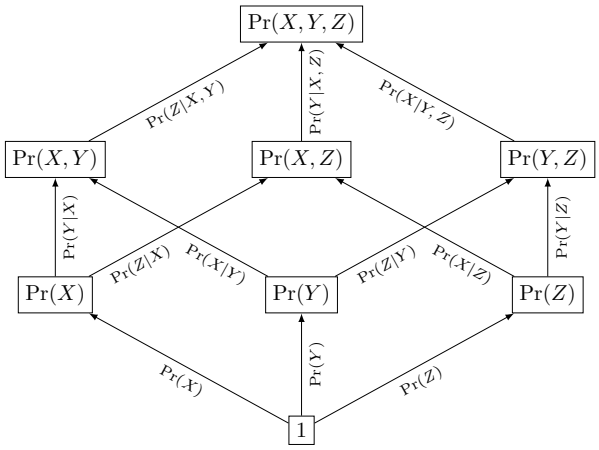


Figure 1. Factorization Lattice of $\Pr(X, Y, Z)$

We propose a new estimation algorithm for MCAR data, called *factored deletion*, which aggregates the estimates from all possible factorizations of $\Pr(\mathbf{Y})$. The number of such factorizations is $k!$, where k is the number of variables in a family $\mathbf{Y}$. However, different factorizations can share the same sub-factors, which can be estimated once and reused across factorizations. These computations can be organized using a lattice, as illustrated in Figure 1, which has only 2^k nodes and $k2^{k-1}$ edges. Our algorithm will compute as many estimates as there are edges in this lattice, which is only on the order of $O(n \log n)$, where n is the number of parameters being estimated for a family Y (which is also exponential in the number of variables k).

More specifically, our factored deletion algorithm operates as follows. First, we estimate the conditional probabilities on the edges of the lattice, each estimate using the subset of the data where its variables are observed. Second, we propagate the estimates bottom-up through the lattice. For each node, there may be several alternative estimates available, on its incoming edges. To aggregate these estimates, we propose to use an inverse-variance weighting heuristic. Note that whereas direct deletion uses only those instances in the data where *all* variables in $\mathbf{Y}$ are observed, *factored deletion* can use any instance in the data where *at least one* variable in $\mathbf{Y}$ is observed.

Factored Deletion: MAR Using factored deletion algorithm, we shall now derive an estimand for family marginals $P(\mathbf{Y})$ under the MAR assumption. Let $\mathbf{X}'_o =$

$\mathbf{X}_o \setminus \mathbf{Y}_o$ denote the fully-observed variables outside of the family variables $\mathbf{Y}$ (and thus $\mathbf{X}_o = \mathbf{Y}_o \cup \mathbf{X}'_o$). Let $Y_1, Y_2, ...Y_{|\mathbf{Y}_m|}$ be an ordering of the partially observed variables in $\mathbf{Y}$. We then have:

$$
\begin{aligned}
&\Pr(\mathbf{Y}) \\
&= \sum_{\mathbf{X}'_o} \Pr(\mathbf{X}'_o) \prod_i \Pr(Y_i | Z, \mathbf{X}'_o) \text{ where } Z = \bigcup_{j=i+1}^{|\mathbf{Y}_m|} \mathbf{Y}_j \\
&= \sum_{\mathbf{X}'_o} \Pr(\mathbf{X}'_o) \prod_i \Pr(Y_i | Z, R_{Z'} = \mathsf{ob}, \mathbf{X}'_o) \\
&\quad \text{where } Z' = \mathbf{Y}_m \cap (Z \cup \{Y_i\}) \\
&= \sum_{\mathbf{X}'_o} \Pr(\mathbf{X}'_o) \prod_i \Pr(Y_i^{'} | Z^{\star}_m, Z_o, R_{Z'} = \mathsf{ob}, \mathbf{X}'_o) \\
&\quad \text{where } Y_i^{'} = Y_i^{\star} \text{ if } Y_i \in \mathbf{X}_m \text{ and } Y_i^{'} = Y_i \text{ otherwise.} \\
&\approx \sum_{\mathbf{X}'_o} \Pr_{\mathfrak{D}}(\mathbf{X}'_o) \prod_i \Pr_{\mathfrak{D}}(Y_i^{'} | Z^{\star}_m, Z_o, R_{Z'} = \mathsf{ob}, \mathbf{X}'_o)
\end{aligned}
$$

4. Empirical Evaluation

To evaluate the proposed learning algorithms, we generate partially observed datasets from Bayesian networks, and relearn their parameters from the data.[2] Our empirical comparison involves the following algorithms:

EM-k Expectation maximization using the jointree inference algorithm and k random restarts.

D-MCAR Direct deletion for MCAR data (Section 3.1).

F-MCAR Factored deletion for MCAR data (Section 3.3).

D-MAR Direct deletion for MAR data (Section 3.2).

F-MAR Factored deletion for MAR data (Section 3.2).

L+EM Expectation maximization using the jointree inference algorithm, seeded by the estimates of learner L.

Algorithms D-MCAR and F-MCAR are consistent for MCAR data only, while D-MAR and F-MAR are consistent for general MAR data. EM is consistent for MAR data if it converges to maximum likelihood estimates.

Given the multitude of Bayesian networks in use today, and the different types of missing data mechanisms one can conceive of, it is not feasible to provide a conclusive practical comparison of these algorithms for all learning problems. Instead, we choose to report on individual experiments that we find representative of general trends, and that highlight trade-offs between the learning algorithms. For a more exhaustive comparison, with six different Bayesian networks and many different missing data mechanisms, we refer the reader to the supplementary online material.

[2]The implementation and experimental setup will be made available online.

Size	EM-1	EM-10	D-MCAR	F-MCAR	D-MAR
Runtime [s]					
10^2	0	5	0	0	0
10^3	6	58	0	0	0
10^4	76	-	0	2	0
10^5	-	-	2	24	4
10^6	-	-	19	197	20
Kullback-Leibler Divergence					
10^2	2.278	2.278	2.715	2.443	3.786
10^3	0.347	0.346	0.585	0.473	0.742
10^4	0.045	-	0.119	0.087	0.178
10^5	-	-	0.016	0.012	0.047
10^6	-	-	0.002	0.001	0.013
Test Set Log-Likelihood (Fully Observed)					
10^2	-12.05	-12.05	-12.44	-12.17	-13.54
10^3	-10.37	-10.37	-10.60	-10.49	-10.77
10^4	-10.01	-	-10.08	-10.05	-10.14
10^5	-	-	-9.98	-9.97	-10.01
10^6	-	-	-9.97	-9.96	-9.97

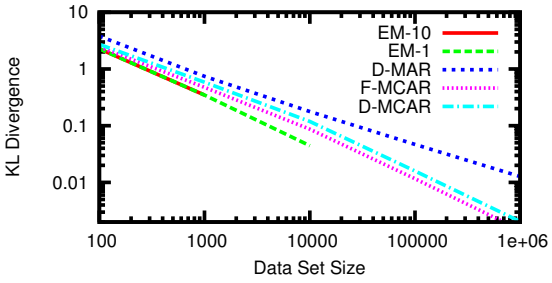
Table 1. Alarm network with MCAR data

We evaluate the learned parameters in terms of their *likelihood* on independently generated, fully-observed test data, and the *Kullback-Leibler divergence* (KLD) between the original and learned Bayesian network. We reported per-instance log-likelihoods (which are divided by dataset size). The supplementary material further shows likelihoods on the training data, before and after data was made missing. We evaluate the learned models on unseen data, so all learning algorithms assume a symmetric Dirichlet prior on the Bayesian network parameters with a concentration parameter of 2. All reported numbers are averaged over 30 repetitions with different random learning problems. When no number is reported, a 5 minute time limit was exceeded.

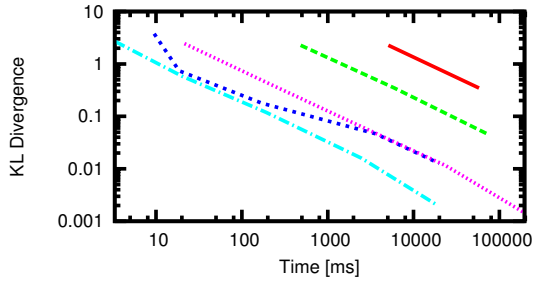
4.1. MCAR Data

We first investigate learning from *MCAR data*, by generating MCAR datasets of increasing size, and evaluating the quality of the learned parameters for each algorithm. Table 1 shows results for the "Alarm" Bayesian network, which has 37 variables. Each training set is generated by sampling from the original Bayesian network, selecting 30% of the variables to be partially observed, and removing 70% of their values completely at random.

We first note that there is no advantage in running EM with restarts: EM-1 and EM-10 learn almost identical models. This indicates that the likelihood landscape for MCAR data has few local optima, and is easy to optimize. Direct and factored deletion are orders of magnitude faster than EM,



(a) KL Divergence vs. Data Size



(b) KL Divergence vs. Time

Figure 2. Runtime behavior with MCAR data, as show in Table 1

which needs to repeatedly run inference for every instance in the dataset. Even though EM outperforms F-MCAR in terms of KLD and likelihood, the difference is negligible, in the sense that only a small difference in the amount of available data makes F-MCAR outperform EM. F-MCAR is slower than D-MCAR, because it requires estimating more probabilities (one for each lattice edge). F-MCAR does learn better models, because it can use a larger portion of the available data. Finally, D-MAR performs worse than F-MCAR and D-MCAR, as it is operating on the weaker MAR assumption. All learners are consistent, as can be seen from the KLD converging to zero.

To illustrate the trade-off between data and computational resources, Figure 2 shows the KLDs from Table 1 as a function of dataset size and time. When data is limited, and computation power is abundant, it is clear that EM is the algorithm of choice, even though the differences are small. When computation power is limited (e.g., when the Bayesian network is highly intractable), and data is abundant (e.g., the online learning or big data setting), the differences are marked. EM is several orders of magnitudes slower than D-MCAR at learning a model of similar quality. F-MCAR may provide a good trade-off.

4.2. MAR Data

We now investigate the more challenging problem of learning from *MAR data*, which are generated as follows: (a) select $m\%$ of the variables to be partially observed, (b) introduce a missingness mechanism variable R_X for each partially observed variable X, (c) each R_X gets assigned p parents that are randomly selected from the set of observed variables, giving preference to neighbors of X in the Bayesian network, (d) sample parameters for the missingness mechanism CPTs from a Beta distribution, and (e) sample a complete dataset, then sample R_X values, and hide X values accordingly.

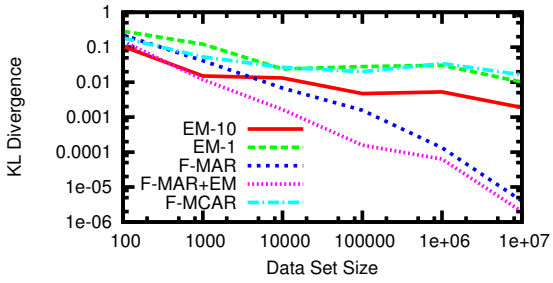
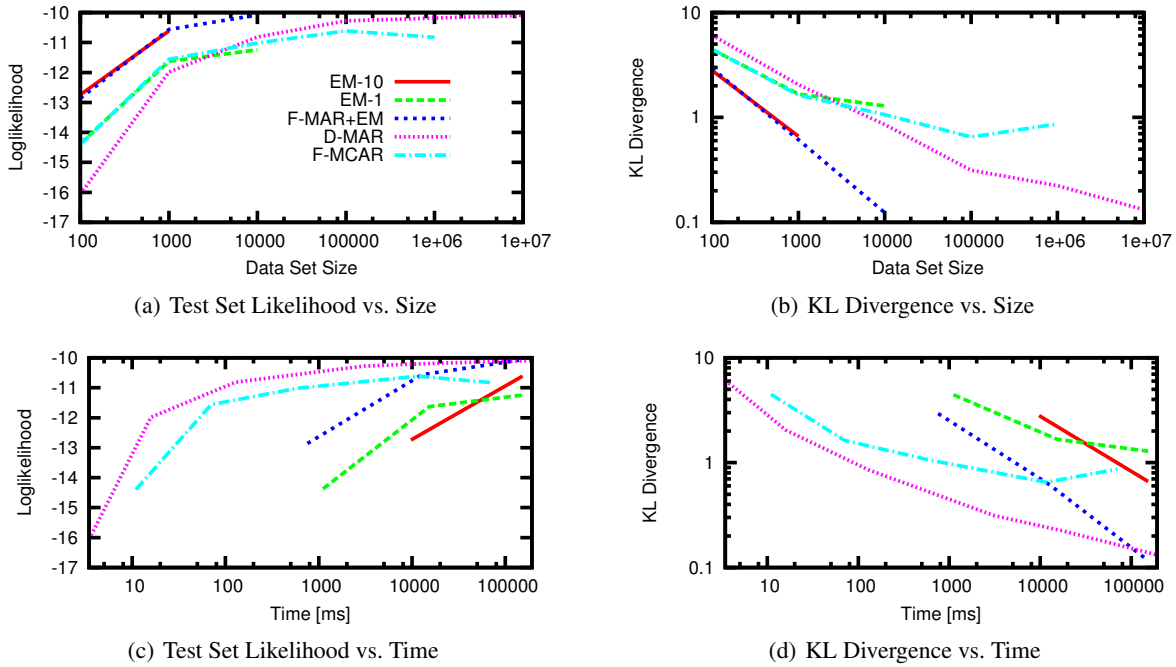


Figure 3. Fire Alarm network with MAR data.

For our first MAR experiment, we work with a small network that is tractable enough for EM to scale to large dataset sizes. Figure 3 shows KLD for the "Fire Alarm" network, which has only 6 variables. The missing data mechanisms has $m = 0.3$, $p = 2$, and a Beta distribution with shape parameters 1.0 and 0.5.

Now, there is a large difference between EM with and without restarts, indicating that the likelihood landscape is much more challenging to optimize. EM with 10 restarts performs well for small dataset sizes, but stops converging after seeing around 1,000 instances. This can be due to all restarts of EM getting stuck in local optima. The KLD of factored deletion for MAR starts off between EM and EM-10 for small sizes, but quickly outperforms EM. For very large dataset sizes, it learns networks whose KLD is several orders of magnitude smaller than EM. The highest-quality models are obtained from combining F-MAR with EM, by providing the F-MAR learned parameters as seeds for EM. This approach is on par with EM-10 for small datasets, while still converging for large dataset sizes. F-MCAR is theoretically not consistent for MAR data, and therefore converges to a biased estimate whose KLD is around 0.01. Even though EM is theoretically consistent on MAR data, it is not so in practice, as it converges to a biased estimate that is similar to F-MCAR.

For our second MAR experiment, we work with the less tractable "Alarm" Bayesian network, using the same missing data mechanisms as in our first MAR experiment. Figure 4 shows test set likelihoods and KL divergences, as a function of both dataset size and time. EM-10 again outperforms EM, but since inference is more challenging in "Alarm", EM-10 fails to scale beyond 1,000 instances,



(a) Test Set Likelihood vs. Size

(b) KL Divergence vs. Size

(c) Test Set Likelihood vs. Time

(d) KL Divergence vs. Time

Figure 4. Alarm network with MAR data.

whereas EM-1 scales to 10,000. EM-1 outperforms D-MAR with less than 2,000 instances, but loses to D-MAR with more instances. Again, EM seeded by F-MAR strikes a balance, achieving the same quality as EM-10, with the run time cost of EM-1. D-MAR dominates all versions of EM as a function of runtime. F-MCAR performs well with little data, but converges to a biased estimate.

5. Learning with a Missingness Graph

In the previous section, we made very general assumptions about the structure of the missingness graph, which captures MCAR and MAR datasets. In this section, we show that we can exploit additional knowledge about the structure of the missingness graph, so that we can take advantage of more data, to obtain more accurate parameter estimates more efficiently.

5.1. Informed Deletion

To compute our parameter estimates for MAR data, corresponding to Equation 1, we sum over all variable instantiations $\mathbf{x}'_o$ of the observed variables $\mathbf{X}'_o$ that lie outside our family $\mathbf{Y}$ (or more precisely, those instantiations $\mathbf{x}'_o$ that appear in the dataset). Given more information about the nature of the missingness mechanisms $\mathbf{R}_{\mathbf{Y}_m}$ (namely, the variables that they depend on), we can both improve the efficiency of the estimation, as well as increase the amount of data available to compute it.

Suppose for now that we know the parents of the missingness mechanisms $\mathbf{R}_{\mathbf{Y}_m}$, denoted by $\mathrm{Pa}(\mathbf{R}_{\mathbf{Y}_m})$ (we will revisit this assumption later). Under the MAR assumption, no parent of $\mathbf{R}_{\mathbf{Y}_m}$ can be a partially-observed variable (otherwise a partially observed variable would not be independent of the missingness mechanisms). Moreover, mechanisms $\mathbf{R}$ do not in general act as parents of the original variables $\mathbf{X} = \mathbf{X}_o \cup \mathbf{X}_m$. Hence, by the Markov property of Bayesian networks (a variable is independent of its non-descendants, given its parents), the mechanisms $\mathbf{R}_{\mathbf{Y}_m}$ are independent of the variables in $(\mathbf{X}_o \cup \mathbf{X}_m) \setminus \mathrm{Pa}(\mathbf{R}_{\mathbf{Y}_m})$ given $\mathrm{Pa}(\mathbf{R}_{\mathbf{Y}_m})$. Let $\mathbf{Z}_o$ denote the set of variables in $\mathrm{Pa}(\mathbf{R}_{\mathbf{Y}_m})$ that are not in $\mathbf{Y}_o$. We can thus reduce the scope of the summation of Equation 1, to just the variables $\mathbf{Z}_o$:

$$\Pr(\mathbf{Y}) \approx \sum_{\mathbf{Z}_o} \Pr_{\mathcal{D}}(\mathbf{Y}^{\star}_m | \mathbf{Y}_o, \mathbf{Z}_o, \mathbf{R}_{\mathbf{Y}_m} = \mathsf{ob}) \Pr_{\mathcal{D}}(\mathbf{Y}_o, \mathbf{Z}_o)$$

Again, reducing the scope of the summation not only allows us to compute this estimate more efficiently, but further allows us to use more data. Moreover, our estimates continue to be consistent. We refer to this estimation algorithm as *informed deletion*.

Note that knowing the parents of a mechanism variable $R \in \mathbf{R}$, is effectively equivalent, for the purposes of informed deletion, to knowing the Markov blanket of R (Pearl, 1987), which can be learned from the data (Yaramakala & Margaritis, 2005; Tsamardinos et al., 2003). With sufficient domain knowledge, an expert may be able to specify the parents of the mechanism variables. It suf-

fices even to identify a subset of the observed variables, that just *contain* the Markov blanket; this knowledge can still be exploited to reduce the scope of the summation. As we shall discuss next, having deeper knowledge of the nature of the missingness mechanisms, will enable us to obtain consistent estimators, even for datasets that are not MAR (in some cases).

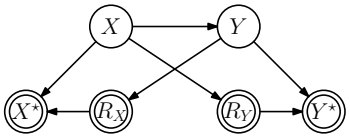
5.2. Empirical Evaluation

For our final experiment, we evaluate the benefits of informed deletion. In addition to the MAR assumption, with this setting, we assume that we know the set of parents of the missingness mechanism variables $\mathbf{Z}_o$.

To generate data that follows such a mechanisms, we select a random set of s variables in $\mathbf{Z}_o$. We further employ the sampling algorithm previously used for MAR data, but now insist that the parents of R variables come from $\mathbf{Z}_o$. Table 2 shows likelihoods and KLDs on the Alarm network, for $s = 3$, and other settings as in the MAR experiments. Informed D-MAR (ID-MAR) and F-MAR (IF-MAR) consistently outperform their non-informed counterparts.

5.3. Missing Not at Random (MNAR)

A missing data problem that is neither MCAR nor MAR is classified as *Missing Not at Random* (MNAR). In this case, the parameters of a Bayesian network may not even be identifiable. Further, maximum likelihood estimation is in general not consistent, so the EM algorithm and gradient methods are expected to yield biased estimates. However, if one knows the interactions of the mechanisms that dictate the missingness of a dataset (in the form of a missingness graph), then it becomes possible again to obtain consistent estimates, at least in some cases (Mohan et al., 2013). For example, consider the following missingness graph:



which is an MNAR problem, where both variables X and Y are partially observed, and the missingness of each variable depends on the value of the other. In this case, it is still possible to obtain consistent parameter estimates:

$$\Pr(X,Y) = \frac{\Pr(R_X=\mathsf{ob},R_Y=\mathsf{ob})\Pr(X^\star,Y^\star|R_X=\mathsf{ob},R_Y=\mathsf{ob})}{\Pr(R_X=\mathsf{ob}|Y^\star,R_Y=\mathsf{ob})\Pr(R_Y=\mathsf{ob}|X^\star,R_X=\mathsf{ob})}$$

For a derivation, see (Mohan et al., 2013). Such derivations for recovering queries under MNAR are extremely sensitive to the structure of the missingness graph. Indeed, the class of missingness graphs that admit consistent estimation has not yet been fully characterized. We view, as interesting future work, the identification of missingness graph structures that guarantee consistent estimators (beyond MCAR and MAR), under minimal assumptions (such as the ones we exploited for informed deletion).

Size	F-MCAR	D-MAR	F-MAR	ID-MAR	IF-MAR
Kullback-Leibler Divergence					
10^2	1.921	2.365	2.364	2.021	2.011
10^3	0.380	0.454	0.452	0.399	0.375
10^4	0.073	0.071	0.072	0.059	0.053
10^5	0.041	0.021	0.022	0.011	0.010
10^6	0.040	0.006	0.008	0.001	0.001
Test Set Log-Likelihood (Fully Observed)					
10^2	-11.67	-12.13	-12.13	-11.77	-11.76
10^3	-10.40	-10.47	-10.47	-10.42	-10.40
10^4	-10.04	-10.04	-10.04	-10.02	-10.02
10^5	-10.00	-9.98	-9.98	-9.97	-9.97
10^6	-10.00	-9.97	-9.97	-9.96	-9.96

Table 2. Alarm network with Informed MAR data

6. Related Work

For parameter estimation in Bayesian networks, maximum likelihood estimation is the typical approach used, where for incomplete data, algorithms such as Expectation-Maximization (EM) and gradient methods are typically employed (Dempster et al., 1977; Lauritzen, 1995); see also, e.g., (Darwiche, 2009; Koller & Friedman, 2009; Murphy, 2012; Barber, 2012). As we discussed earlier, such methods do not scale well as they (1) are iterative, (2) require inference in a Bayesian network, and (3) suffer from local optima. Considerable effort has been expended in improving on EM, across multiple dimensions. For example, much work has been devoted to find ways to (1) accelerate the convergence of EM, and to intelligently sample subsets of a dataset; see, e.g., (Thiesson et al., 2001), (2) use approximate inference algorithms in lieu of exact ones when inference is intractable; see, e.g., (Ghahramani & Jordan, 1997; Caffo et al., 2005), and (3) escape local optima; see, e.g., (Elidan et al., 2002).

In the case of complete data, the parameter estimation task simplifies considerably, in the case of Bayesian networks: maximum likelihood estimates can be obtained inference-free and in closed-form, using just a single pass over the data: $\theta_{x|\mathbf{u}} = Pr_{\mathcal{D}}(x|\mathbf{u})$. In fact, the estimation algorithms that we proposed in this paper also obtain the same parameter estimates in the case of complete data, although we are not concerned with maximum likelihood estimation here—we simply want to obtain estimates that are consistent.

Other inference-free estimators have also been proposed

for other classes of probabilistic graphical models. (Abbeel et al., 2006) identified a method for closed-form, inference-free parameter estimation in factor graphs of bounded degree from complete data. More recently, (Halpern & Sontag, 2013) proposed an efficient, inference-free method for consistently estimating the parameters of noisy-or networks from data with latent variables, under certain structural assumptions. We note that inference-free learning of the parameters of: Bayesian networks under MAR data (this paper), factor graphs of bounded degree, under complete data (Abbeel et al., 2006), and structured noisy-or Bayesian networks, with latent variables (Halpern & Sontag, 2013), are all surprising results. From the perspective of maximum likelihood learning, where evaluating the likelihood (requiring inference) seems to be unavoidable, the ability to consistently estimate parameters without the need for inference, greatly extends the accessibility and potential of such models. Without doubt, the ability to estimate Bayesian networks under complete data, in closed-form without inference, has contributed to their broad adoption across numerous domains.

7. Conclusion

When learning Bayesian network parameters under incomplete datasets, where variables are MCAR or MAR, the common wisdom among machine learning practioners is that one needs to use Expectation-Maximization (EM) or gradient methods. However, such methods do not scale well to large datasets or complex Bayesian networks, as they are iterative, they require inference in a potentially intractable network, and they can suffer from local optima. In this paper, we proposed an inference-free, closed-form method for consistently learning Bayesian network parameters, from MCAR and MAR datasets. We further introduced and discussed improved approaches for parameter estimation, when given additional knowledge of the missingness mechanisms underlying an incomplete dataset. Empirically, we demonstrate the practicality of our method, showing that it is orders-of-magnitude more efficient than EM, allowing it to scale to much larger datasets. Further, given access to enough data, we show that our method can learn much more accurate Bayesian networks as well.

References

Abbeel, Pieter, Koller, Daphne, and Ng, Andrew Y. Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7:1743–1788, 2006.

Barber, David. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

Caffo, Brian S., Jank, Wolfgang, and Jones, Galin L. Ascent-based monte carlo expectation-maximization. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):pp. 235–251, 2005.

Chavira, Mark and Darwiche, Adnan. Encoding CNFs to empower component analysis. In *Proceedings of SAT*, pp. 61–74, 2006.

Chavira, Mark and Darwiche, Adnan. Compiling Bayesian networks using variable elimination. In *Proceedings of IJCAI*, pp. 2443–2449, 2007.

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

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

Elidan, Gal, Ninio, Matan, Friedman, Nir, and Shuurmans, Dale. Data perturbation for escaping local maxima in learning. In *Proceedings of AAAI*, pp. 132–139, 2002.

Ghahramani, Zoubin and Jordan, Michael I. Factorial hidden markov models. *Machine Learning*, 29(2-3):245–273, 1997.

Halpern, Yoni and Sontag, David. Unsupervised learning of noisy-or Bayesian networks. In *Proceedings of UAI*, 2013.

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

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

Mohan, Karthika, Pearl, Judea, and Tian, Jin. Graphical models for inference with missing data. In *Proceedings of NIPS*, 2013.

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

Pearl, Judea. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32(2):245–257, 1987.

Rubin, Donald B. Inference and missing data. *Biometrika*, 63(3): 581–592, 1976.

Thiesson, Bo, Meek, Christopher, and Heckerman, David. Accelerating EM for large databases. *Machine Learning*, 45(3): 279–299, 2001.

Tsamardinos, Ioannis, Aliferis, Constantin F, Statnikov, Alexander R, and Statnikov, Er. Algorithms for large scale Markov blanket discovery. In *Proceedings of FLAIRS*, volume 2003, pp. 376–381, 2003.

Yaramakala, Sandeep and Margaritis, Dimitris. Speculative markov blanket discovery for optimal feature selection. In *Proceedings of ICDM*, 2005.