Bayesian Network Parameter Learning using EM with Parameter Sharing

Erik Reed
Electrical and Computer Engineering
Carnegie Mellon University

Ole J. Mengshoel
Electrical and Computer Engineering
Carnegie Mellon University

Abstract

This paper explores the effects of parameter sharing on Bayesian network (BN) parameter learning when there is incomplete data. Using the Expectation Maximization (EM) algorithm, we investigate how varying degrees of parameter sharing, varying number of hidden nodes, and different dataset sizes impact EM performance. The specific metrics of EM performance examined are: likelihood, error, and the number of iterations required for convergence. These metrics are important in a number of applications, and we emphasize learning of BNs for diagnosis of electrical power systems. One main point, which we investigate both analytically and empirically, is how parameter sharing impacts the error associated with EM's parameter estimates.

1 INTRODUCTION

Bayesian network (BN) conditional probability tables (CPTs) can be learned when the BN structure is known, for either complete or incomplete data. Different algorithms have been explored in the case of incomplete data, including: Expectation Maximization [8, 14, 15, 28], Markov Chain Monte Carlo methods such as Gibbs sampling [17], and gradient descent methods [9]. Expectation Maximization (EM) seeks to maximize the likelihood, or the Maximum a Posteriori (MAP) estimate, for the BN CPTs.

We focus in this paper on EM [8, 14, 15, 28], an iterative algorithm that converges to a maximum likelihood estimate (MLE). While EM is powerful and popular, there are several challenges that motivate our research. First, when computing MLEs, EM is easily trapped in local optima and is typically very sensitive to the placement of initial CPT values. Methods of making EM less prone to getting trapped in local optimal have been investigated [11, 18, 34, 38]. Second, EM is often computationally demanding, especially when the BN is complex and there is much data [2, 3, 29, 35]. Third, parameters that EM converges to can be far from the true probability distribution, yet still have a high likelihood. This is a limitation of EM based on MLE.

In this paper we investigate, for known BN structures, how varying degree of parameter sharing [17, 25, 26], varying number of hidden nodes, and different dataset sizes impact EM performance. Specifically, we are:

- running many random initializations (or random restarts) of EM, a technique known to effectively counter-act premature convergence [10, 22];
- recording for each EM run the following metrics: (i) log-likelihood ($\ell\ell$) of estimated BN parameters, (ii) error (the Euclidean distance between true and estimated BN parameters), and (iii) number of EM iterations until convergence; and
- testing BNs with great potential for parameter sharing, with a focus on electrical power system BNs (reflecting electrical power system components known to exhibit similar behavior).

Even when EM converges to a high-likelihood MLE, the error can be large and vary depending on initial conditions. This is a fundamental limitation of EM using MLE; even a BN with high likelihood may be far from the true distribution and thus have a large error. Error as a metric for the EM algorithm for BN parameter learning has not been discussed extensively in the existing literature. The analysis and experiments in this paper provide new insights in this area.

Our main application is electrical power systems, and in particular NASA's Advanced Diagnostics and Prognostics Testbed (ADAPT) [27]. ADAPT has already been represented as BNs, which have proven themselves as very well-suited to electrical power system

health management [12, 19–21, 30–33]. Through compilation of BNs to arithmetic circuits [4, 5], a broad range of discrete and continuous faults can be detected and diagnosed in a computationally efficient and predictable manner, resulting in award-winning performance in international diagnostic competitions [30].[1] From a machine learning and EM perspective, as considered in this paper, it is hypothesized that the learning of ADAPT BNs may benefit from parameter sharing. This is because there are several repeated BN nodes and fragments in these BNs. In addition to parameter sharing, we study in this paper the impact on EM of varying the number of hidden nodes, reflecting different sensing capabilities.

Why are BNs and arithmetic circuits useful for electrical power system diagnostics? First, power systems exhibit multi-variate uncertainty, for example regarding component and sensor health (are they working or failing?) as well as noisy sensor readings. Second, there is substantial local structure, as reflected in an EPS schematic, that can be taken advantage of when constructing a BN automatically or semi-automatically [19,20,30]. Consequently, BN treewidth is small enough for exact computation using junction trees, variable elimination, or arithmetic circuits to be feasible [19,30]. Third, we compile BNs into arithmetic circuits [4, 5], which are fast and predictable in addition to being exact. These are all important benefits in cyber-physical systems including electrical power systems.

The rest of this paper is structured as follows. In Section 2, we introduce BNs, parameter learning for incomplete data using EM, and related research. Section 3 presents our main application area, electrical power systems. In Section 4, we define the sharing concept, discuss sharing in EM for BN parameter learning, and provide analytical results. In Section 5 we present experimental results for parameter sharing in BNs when using EM, emphasizing electrical power system fault diagnosis using BNs. Finally, we conclude and outline future research opportunities in Section 6.

Notation	Explanation
$\boldsymbol{X}$	BN nodes
$\boldsymbol{W}$	BN edges
$\boldsymbol{\theta}$	BN CPTs
$\hat{\boldsymbol{\theta}}$	estimated CPTs
$\boldsymbol{\theta}^*$	true CPTs
$\boldsymbol{O}$	observable nodes
$\boldsymbol{H}$	hidden nodes
$\boldsymbol{S}$	(actually) shared nodes
$\boldsymbol{P}$	(potentially) shared nodes
$\boldsymbol{U}$	unshared nodes
$\boldsymbol{Y}$	set partition of $\boldsymbol{X}$
$T_{\boldsymbol{P}}$	number of wrong CPTs
$\boldsymbol{E}$	evidence nodes
$\boldsymbol{R}$	non-evidence nodes
$t_{\min}$	min # of EM iterations
$t_{\max}$	max # of EM iterations
t'	iteration # at EM convergence
ϵ	tolerance for EM
$err(\hat{\boldsymbol{\theta}})$	error of $\hat{\boldsymbol{\theta}}$ relative to $\boldsymbol{\theta}^*$
$n_{\boldsymbol{A}} = \lvert\boldsymbol{A}\rvert$	cardinality of the set $\boldsymbol{A}$
ℓ	likelihood
$\ell\ell$	log-likelihood
$\beta = (\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta})$	Bayesian network (BN)
$E(Z)$	expectation of r.v. Z
$V(Z)$	variance of r.v. Z
r	Pearson's corr. coeff.
$\theta \in [0, 1]$	CPT parameter
$\hat{\theta} \in [0, 1]$	estimated CPT parameter
$\theta^* \in [0, 1]$	true CPT parameter
δ	error bound for θ

Table 1: Notation used in this paper.

2 BACKGROUND

This section presents preliminaries including notation (see also Table 1).

2.1 BAYESIAN NETWORKS

Consider a BN $\beta = (\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta})$, where $\boldsymbol{X}$ are discrete nodes, $\boldsymbol{W}$ are edges, and $\boldsymbol{\theta}$ are CPT parameters. Let $\boldsymbol{E} \subseteq \boldsymbol{X}$ be evidence nodes, and $\boldsymbol{e}$ the evidence. A BN factors a joint distribution $\Pr(\boldsymbol{X})$, enabling different probabilistic queries to be answered by efficient algorithms; they assume that nodes $\boldsymbol{E}$ are clamped to values $\boldsymbol{e}$. One query of interest is to compute a most probable explanation (MPE) over the remaining nodes $\boldsymbol{R} = \boldsymbol{X} \setminus \boldsymbol{E}$, or $\text{MPE}(\boldsymbol{e})$. Computation of marginals (or beliefs) amounts to inferring the posterior probabilities over one or more query nodes $\boldsymbol{Q} \subseteq \boldsymbol{R}$, specifically $\text{BEL}(Q, \boldsymbol{e})$, where $Q \in \boldsymbol{Q}$.

In this paper, we focus on situations where $\boldsymbol{\theta}$ needs to be estimated but the BN structure ($\boldsymbol{X}$ and $\boldsymbol{W}$) is known. Data is complete or incomplete; in other words there may be hidden nodes $\boldsymbol{H}$ where $\boldsymbol{H} = \boldsymbol{X} \setminus \boldsymbol{O}$ and $\boldsymbol{O}$ are observed. A dataset is defined as $(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ with m samples (observations), where $\mathbf{x}_i$ is a vector of instantiations of nodes $\boldsymbol{X}$ in the complete data case. When the data is complete, the BN parameters $\boldsymbol{\theta}$ are often estimated to maximize the data likelihood (MLE). In this paper, for a given dataset, a variable $X \in \boldsymbol{X}$ is either observable ($X \in \boldsymbol{O}$) or hidden ($X \in \boldsymbol{H}$); it is not hidden for just a strict subset of the samples.[2] Let $\boldsymbol{H} > 0$. For each hidden node

[1] Further information can be found here: `https://sites.google.com/site/dxcompetition/`.

[2] In other words, a variable that is completely hidden in the training data is a latent variable. Consequently, its

$H \in \boldsymbol{H}$ there is then a "?" or "N/A" in each sample. Learning from incomplete data also relies on a likelihood function, similar to the complete data case. However, for incomplete data several properties of the complete data likelihood function–such as unimodality, a closed-form representation, and decomposition into a product form–are lost. As a consequence, the computational issues associated with BN parameter learning are more complex, as we now discuss.

2.2 TRADITIONAL EM

For the problem of optimizing such multi-dimensional, highly non-linear, and multimodal functions, several algorithms have been developed. They include EM, our focus in this paper. EM performs a type of hill-climbing in which an estimate in the form of an expected likelihood function ℓ is used in place of the true likelihood ℓ.

Specifically, we examine the EM approach to learn BN parameters $\boldsymbol{\theta}$ from incomplete data sets.[3] The *traditional EM* algorithm, without sharing, initializes parameters to $\boldsymbol{\theta}^{(0)}$. Then, EM alternates between an E-step and an M-step. In the t-th E-step, using parameters $\boldsymbol{\theta}^{(t)}$ and observables from the dataset, EM generates the likelihood $\ell^{(t)}$ taking into account the hidden nodes $\boldsymbol{H}$. In the M-step, EM modifies the parameters to $\boldsymbol{\theta}^{(t+1)}$ to maximize the data likelihood. While $|\ell^{(t)} - \ell^{(t-1)}| \geq \epsilon$, where ϵ is a tolerance, EM repeats from the E-step.

EM monotonically increases the likelihood function ℓ or the log-likelihood function $\ell\ell$, thus EM converges to a point $\hat{\boldsymbol{\theta}}$ or a set of points (a region). Since $\ell\ell$ is bounded, EM is guaranteed to converge. Typically, EM converges to a local maximum [37] at some iteration t', bounded as follows: $t_{\max} \geq t' \geq t_{\min}$. Due to the use of ϵ above, it is for practical implementations of EM with restart more precise to discuss regions of convergence, even when there is point-convergence in theory.

One topic that has been discussed is the initialization phase of the EM algorithm [8]. A second research topic is stochastic variants of EM, typically known as Stochastic EM [7,11]. Generally, Stochastic EM is concerned with improving the computational efficiency of EM's E-step. Several other methods for increasing the efficacy of the EM algorithm for BNs exist. These include parameter constraints [1, 6, 25], parameter inequalities [26], exploiting domain knowledge [17, 24], and parameter sharing [13, 17, 25, 26].

When data is incomplete, the BN parameter estimation problem is in general non-identifiable. There may be several parameter estimates $\hat{\boldsymbol{\theta}}_1$, ..., $\hat{\boldsymbol{\theta}}_m$ that have the same likelihood, given the dataset [36]. Thus, we need to be careful when applying standard asymptotic theory from statistics (which assumes identifiability) and when interpreting a learned model. Section 4.2 introduces an error measure that provides some insight regarding identifiability, since it measures distance from the true distribution $\boldsymbol{\theta}^*$.

3 ELECTRICAL POWER SYSTEMS

Electrical Power Systems (EPSs) are critical in today's society, for instance they are essential for the safe operation of aircraft and spacecraft. The terrestrial power grid's transition into a smart grid is also very important, and the emergence of electrical power in hybrid and all-electric cars is a striking trend in the automotive industry.

ADAPT (Advanced Diagnostics and Prognostics Testbed) is an EPS testbed developed at NASA [27]. Publicly available data from ADAPT is being used to develop, evaluate, and mature diagnosis and prognosis algorithms. The EPS functions of ADAPT are as follows. For power generation, it currently uses utility power with battery chargers (there are also plans to investigate solar power generation). For power storage, ADAPT contains three sets of 24 VDC 100 Amp-hr sealed lead acid batteries. Power distribution is aided by electromechanical relays, and there are two load banks with AC and DC outputs. For control and monitoring there are two National Instruments compact FieldPoint backplanes. Finally, there are sensors of several types, including for: voltage, current, temperature, light, and relay positions.

ADAPT has been used in different configurations and represented in several fault detection and diagnosis BNs [12, 19–21, 30–33], some of which are investigated in this paper (see Table 2). Each ADAPT BN node typically has two to five discrete states. BN nodes represent, for instance: sensors (measuring, for example, voltage or temperature); components (for example batteries, loads, or relays); or system health of components and sensors (broadly, they can be in "healthy" or "faulty" states).

3.1 SHARED VERSUS UNSHARED

In BN instances where parameters are not shared, the CPT for a node in the BN is treated as separate from the other CPTs. The assumption is not always reasonable, however. ADAPT BNs may benefit from shared parameters, because there are typically several

true distribution is not identifiable.

[3] Using EM, learning from complete data is a special case of learning from incomplete data.

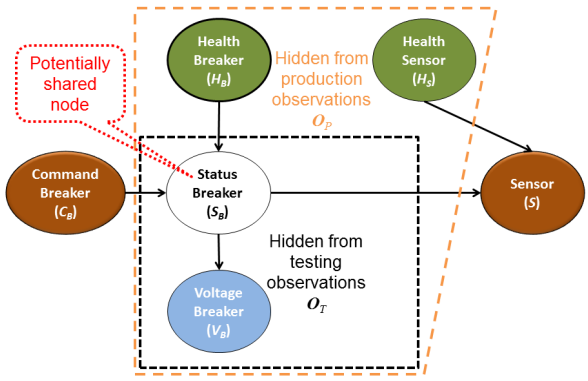


Figure 1: One of three similar sub-networks in the mini-ADAPT BN. Under parameter sharing, the S_B node is shared between the three sub-networks.

repeated nodes or fragments in these BNs [21, 30]. It is reasonable to assume that "identical" power system sensors and components will behave in similar ways. More broadly, sub-networks of identical components should function in a similar way to other "identical" sub-networks in a power system, and such knowledge can be the basis for parameter sharing.

For parameter sharing as investigated in this paper, the CPTs of some nodes are assumed to be approximately equal to the CPTs of different nodes elsewhere in the BN.[4] Data for one set of nodes can be used elsewhere in the BN if the corresponding nodes are shared during EM learning. This is a case of parameter sharing involving the global structure of the BN, where different CPTs are shared, as opposed to parameter sharing within a single CPT [13].

In some ADAPT BNs, one sub-network is essentially duplicated three times, reflecting triple redundancy in ADAPT's power storage and distribution network [27]. Such a six-node sub-network from an ADAPT BN is shown in Figure 1. This sub-network was, in this paper, manually selected for further study of sharing due to this duplication. The sub-network is part of the mini-ADAPT BN used in experiments in Section 5.3. In the mini-ADAPT sharing condition, only the node S_B was shared between all three BN fragments. Generally, we define shared nodes $\boldsymbol{S}$ and unshared nodes $\boldsymbol{U}$, with $\boldsymbol{U} = \boldsymbol{X} \setminus \boldsymbol{S}$.

[4]It is unrealistic to assume that several engineered physical objects, even when it is desired that they are exactly the same, in fact turn out to have exactly the same behavior. A similar argument has been made for object-oriented BNs [14], describing it as "violating the OO assumption." We thus say that CPTs are approximately equal rather than equal. Under sharing, however, we are making the simplifying assumption that shared CPTs are equal.

3.2 OBSERVABLE VERSUS HIDDEN

Consider a complex engineered system, such as an electrical power system. After construction, but before it is put into production, it is typically tested extensively. The sensors used during *testing* lead to one set of observation nodes in the BN, $\boldsymbol{O_T}$. The sensors used during *production* lead to another set of observations in the BN, $\boldsymbol{O_P}$. For reasons including cost, fewer sensors are typically used during production than during testing, thus we assume $\boldsymbol{O_P} \subseteq \boldsymbol{O_T}$.

As an example, in Figure 1 we denote $\boldsymbol{O_P} = \{C_B, S\}$ as *production* observation nodes, and $\boldsymbol{O_T} = \{C_B, S, H_B, H_S\}$ as *testing* observation nodes.

In all our experiments, shared nodes are also hidden, or $\boldsymbol{S} \subseteq \boldsymbol{H}$. Typically, there are hidden nodes that are not necessarily shared, or $\boldsymbol{S} \subset \boldsymbol{H}$. This is, for example, the case for the mini-ADAPT BN as reflected in the sub-network in Figure 1.

4 EM WITH SHARING

4.1 SHARING IN BAYESIAN NETWORKS

Consider a BN $\beta = (\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta})$. A sharing set partition for nodes $\boldsymbol{X}$ is a set partition $\boldsymbol{Y}$ of $\boldsymbol{X}$ with subsets $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_k$, with $\boldsymbol{Y}_i \subseteq \boldsymbol{X}$ and $k \geq 1$. For each $\boldsymbol{Y}_i$ with $k \geq i \geq 1$ the nodes $X \in \boldsymbol{Y}_i$ share a CPT during EM learning as discussed in Section 4.2. We assume that the nodes in $\boldsymbol{Y}_i$ have exactly the same number of states. The same applies to their respective parents in β, leading to each $Y \in \boldsymbol{Y}_i$ having the same number of parent instantiations and exactly the same CPTs.

Traditional non-sharing is a special case of sharing in the following way. We assign each BN node to a separate set partition such that for $\boldsymbol{X} = \{X_1, ..., X_n\}$ we have $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_n$ with $\boldsymbol{Y}_i = \{X_i\}$.

One key research goal is to better understand the behavior of EM as sharing nodes $\boldsymbol{S}$ and observable nodes $\boldsymbol{O}$ vary. We examine three cases: complete data $\boldsymbol{O}_C$ (no hidden nodes); a sensor-rich testing setting with observations $\boldsymbol{O}_T$; and a sensor-poor production setting with observations $\boldsymbol{O}_P$. Understanding the impact of varying observations $\boldsymbol{O}$ is important due to cost and effort associated with observation or sensing.

4.2 SHARING EM

Similar to traditional EM (see Section 2.2), the sharing EM algorithm also takes as input a dataset and estimates a vector of BN parameters $\hat{\boldsymbol{\theta}}$ by iteratively improving $\ell\ell$ until convergence. The main difference of sharing EM compared to traditional EM is that we are now setting some nodes as shared $\boldsymbol{S}$, according to

$\boldsymbol{Y}$. To arrive at the sharing EM algorithm from the traditional EM algorithm, we modify the likelihood function to introduce parameter sharing and combine parameters (see also [13, 17]). When running sharing EM, nodes $\boldsymbol{X}$ are treated as separate for the E-step. There is a slightly modified M-step, using $\boldsymbol{Y}$, to aggregate the shared CPTs and parameters.[5] This is the use of *aggregate sufficient statistics*, which considers sufficient statistics from more than one BN node [13].

Let $\hat{\theta}_{i,j}$ be the j'th estimated probability parameter for BN node $X_i \in \boldsymbol{X}$. We define error of $\hat{\boldsymbol{\theta}}$ for BN $(\boldsymbol{X}, \boldsymbol{W}, \hat{\boldsymbol{\theta}})$ as the L^2 distance from the true probability distribution $\boldsymbol{\theta}^*$ from which data is sampled:

$$err(\hat{\boldsymbol{\theta}}) = \sum_i \sqrt{\sum_j \left(\theta^*_{i,j} - \hat{\theta}_{i,j}\right)^2}. \tag{1}$$

This error is the summation of the Euclidean distance between true and estimated CPT parameters, or the L^2 distance, providing an overall distance metric.

Why do we use Euclidean distance to measure error? One could, after all, argue that this distance metric is poor because it does not agree with likelihood. We use Euclidean distance because we are interested not only in the black box performance of the BN, but also its validity and understandability to a human expert.[6] This is important when an expert needs to evaluate, validate, or refine a BN model, for example a BN for an electrical power system.

4.3 EM'S BEHAVIOR UNDER SHARING

We now provide a simple analysis of certain aspects of traditional EM (TEM) and sharing EM (SEM). For simplicity, we only consider EM runs that converge and exclude runs that time out.[7]

For a node $X_i \in \boldsymbol{X}$, TEM will converge to one among potentially several convergence regions. Suppose that the CPT of node X_i has $\kappa(X_i)$ convergence regions. Then the actual number of convergence regions $\kappa(\beta_U)$ for a non-shared BN β_U with nodes $\boldsymbol{X} = \{X_1, ..., X_n\}$ is upper bounded by $\bar{\kappa}(\beta_U)$ as follows:

$$\kappa(\beta_U) \leq \bar{\kappa}(\beta_U) = \prod_{i=1}^{n} \kappa(X_i). \tag{2}$$

[5]For the relevant LibDAI source code, please see here: `https://github.com/erikreed/HadoopBNEM/blob/master/src/emalg.cpp#L167`. In words, it is a modification on the collection of sufficient statistics during the maximization step.

[6]We assume that (1) is better than likelihood in this regard, if the original BN was manually constructed. The BNs experimented with in Section 5 were manually constructed, for example.

[7]In practice, runs that time out are very rare with the parameter settings we use in experiments.

Due to the sharing, SEM intuitively has fewer convergence regions than TEM. This is due to SEM's slightly modified M-step that aggregates the shared CPTs and parameters. Consider a BN β_S with exactly the same nodes and edges as β_U, but with sharing, specifically with sharing set partitions $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_k$ and $k < n$. Without loss of generality, assume that $X_i \in \boldsymbol{Y}_i$. Then the actual number of convergence regions $\kappa(\beta_S)$ is upper bounded by $\bar{\kappa}(\beta_S)$ as follows:

$$\kappa(\beta_S) \leq \bar{\kappa}(\beta_S) = \prod_{i=1}^{k} \kappa(X_i), \tag{3}$$

assuming that the $\kappa(X_i)$ convergence regions used for X_i in (2) carry over to $\boldsymbol{Y}_i$.

A special case of (3) is when there exists exactly one $\boldsymbol{Y}' \in \boldsymbol{Y}$ such that $|\boldsymbol{Y}'| \geq 2$ while for any $\boldsymbol{Z} \in \boldsymbol{Y} \backslash \boldsymbol{Y}'$ we have $|\boldsymbol{Z}| = 1$. The experiments performed in Section 5 are all for this special case. Specifically, but without loss of generality, let $\boldsymbol{Y}_i = \{X_i\}$ for $1 \leq i < k$ and $\boldsymbol{Y}_k = \{X_k, ..., X_n\}$ with $n_{\boldsymbol{S}} = n - k + 1$ (*i.e.*, $\boldsymbol{S} = \boldsymbol{Y}_k$ has $n_{\boldsymbol{S}}$ sharing nodes). It is illustrative to consider the ratio:

$$\frac{\bar{\kappa}(\beta_U)}{\bar{\kappa}(\beta_S)} = \frac{\prod_{i=1}^{n} \kappa(X_i)}{\prod_{i=1}^{k} \kappa(X_i)} = \prod_{i=k+1}^{n} \kappa(X_i). \tag{4}$$

Here, we assume that X_i has $\kappa(X_i)$ convergence regions in both β_U and β_S and take into account that for shared nodes $\boldsymbol{Y}_k = \boldsymbol{S}$, CPTs are tied together.

The simple analysis above suggests a non-trivial impact of sharing, given the multiplicative effect of the $\kappa(X_i)$'s for $k+1 \leq i \leq n$ in (4). However, since upper bounds are the focus in this analysis, only a partial and conservative picture is painted. The experiments in Section 5—see for example Figure 3, Figure 5, and Figure 6—provide further details.

4.4 ANALYSIS OF ERROR

We now consider the number of erroneous CPTs as estimated by SEM when sharing is varied. Clearly, a CPT parameter is continuous and its EM estimate is extremely unlikely to be equal to the original parameter. Thus we consider here a discrete variable, based on forming an interval in the one-parameter case. Generally, let a discrete BN node $X \in \boldsymbol{X}$ have k states such that $x_i \in \{x_1, ..., x_k\}$. Consider $\theta_{x_i|\boldsymbol{z}} = \Pr(X = x_i \mid \boldsymbol{Z} = \boldsymbol{z})$ for a parent instantiation $\boldsymbol{z}$. We now have an original CPT parameter $\theta_i \in \{\theta_{x_1|\boldsymbol{z}}, ..., \theta_{x_{k-1}|\boldsymbol{z}}\}$ and its EM estimate $\hat{\theta}_i \in \{\hat{\theta}_{x_1|\boldsymbol{z}}, ..., \hat{\theta}_{x_{k-1}|\boldsymbol{z}}\}$. Let us jointly consider the original CPT parameter θ^*_i and its estimate $\hat{\theta}_i$. If $\hat{\theta}_i \in [\theta^*_i - \delta_i, \theta^*_i + \delta_i]$ we count $\hat{\theta}_i$ as correct, and say $\hat{\theta}_i = \theta^*_i$; else it is incorrect or wrong, and we

say $\hat{\theta}_i \neq \theta_i^*$. This analysis clearly carries over to multiple CPT parameters, parent instantiations, and BN nodes. This shows how we go from a continuous (estimated CPT parameters $\hat{\boldsymbol{\theta}}$) to a discrete value (number of wrong or incorrect CPT estimates), where the latter is used in this analysis.

Suppose that up to $n_{\boldsymbol{P}}$ nodes can be shared. Further suppose that $n_{\boldsymbol{S}}$ nodes are actually shared while $n_{\boldsymbol{U}}$ nodes are unshared, with $n_{\boldsymbol{U}} + n_{\boldsymbol{S}} = n_{\boldsymbol{P}}$. Let $T_{\boldsymbol{P}}$ be a random variable representing the total number of wrong or incorrect CPTs, $T_{\boldsymbol{S}}$ the total for shared nodes, and $T_{\boldsymbol{U}}$ the total for unshared nodes. Clearly, we have $T_{\boldsymbol{P}} = T_{\boldsymbol{S}} + T_{\boldsymbol{U}}$.

Let us first consider the expectation $E(T_{\boldsymbol{P}})$. Due to linearity, we have $E(T_{\boldsymbol{P}}) = E(T_{\boldsymbol{S}}) + E(T_{\boldsymbol{U}})$. In the non-shared case, assume for simplicity that errors are iid and follow a Bernoulli distribution, with probability p of error and $(1-p) = q$ of no error.[8] This gives $E(T_{\boldsymbol{U}}) = n_{\boldsymbol{U}}p$, using the fact that a sum of Bernoulli random variables follows a Binomial distribution.[9]

In the shared case, all shared nodes either have an incorrect CPT $\hat{\theta} \neq \theta^*$ or the correct CPT $\hat{\theta} = \theta^*$. Assuming again probabilities p of error[10] and $(1-p) = q$ of no error, and by using the definition of expectation of Binomials we obtain $E(T_{\boldsymbol{S}}) = n_{\boldsymbol{S}}p$.

Substituting into $E(T_{\boldsymbol{P}})$ we get

$$E(T_{\boldsymbol{P}}) = n_{\boldsymbol{U}}p + n_{\boldsymbol{S}}p = n_{\boldsymbol{P}}p. \tag{5}$$

Let us next consider the variance $V(T_{\boldsymbol{P}})$. While variance in general is not linear, we assume linearity for simplicity, and obtain

$$V(T_{\boldsymbol{P}}) = V(T_{\boldsymbol{U}}) + V(T_{\boldsymbol{S}}). \tag{6}$$

In the non-shared case we have again a Binomial distribution, with well-known variance

$$V(T_{\boldsymbol{U}}) = n_{\boldsymbol{U}}p(1-p). \tag{7}$$

In the shared case we use the definition of variance, put $p_1 = (1-p)$, $p_2 = p$, and $\mu = n_{\boldsymbol{S}}p$, and obtain after some simple manipulations:

$$V(T_{\boldsymbol{S}}) = \sum_{i=1}^{2} p_i(X_i - \mu)^2 = n_{\boldsymbol{S}}^2 p(1-p), \tag{8}$$

and by substituting (7) and (8) into (6) we get

$$V(T_{\boldsymbol{P}}) = p(1-p)((n_{\boldsymbol{P}} - n_{\boldsymbol{S}}) + n_{\boldsymbol{S}}^2). \tag{9}$$

In words, (9) tells us that as the number $n_{\boldsymbol{S}}$ of shared nodes increases at the expense of the number of unshared nodes $n_{\boldsymbol{U}}$, variance due to non-shared nodes decreases linearly, but variance due to sharing increases quadratically. The net effect shown in (9) is that variance $V(T_{\boldsymbol{P}})$ of the error *increases* with the number of shared nodes, according to our analysis above. Expectation, on the other hand, remains *constant* (5) regardless of how many nodes are shared. These analytical results have empirical counterparts as discussed in Section 5, see for example the error sub-plot at the bottom of Figure 2.

5 EXPERIMENTS

We now report on EM experiments for several different BNs, using varying degrees of sharing. We also vary the number of hidden nodes and dataset size. We used $\epsilon = 1e^{-3}$ as an EM convergence criterion, meaning that EM stopped at iteration t' when the $\ell\ell$-score changed by a value $\leq \epsilon$ between iterations $t'-1$ and t' for $t_{\max} \geq t' \geq t_{\min}$. In these experiments, $t_{\max} = 100$ and $t_{\min} = 3$.

5.1 METHODS AND DATA

Bayesian networks. Table 2 presents BNs used in the experiments.[11] Except for the BN Pigs, these BNs all represent (parts of) the ADAPT electrical power system (see Section 3). The BN Pigs has the largest number of nodes that can be shared ($n_{\boldsymbol{P}} = 296$), comprising 67% of the entire BN. The largest BN used, in terms of node count, edges, and total CPT size, is ADAPT_T2.

Datasets. Data for EM learning of parameters for these BNs were generated using forward sampling.[12] Each sample in a dataset is a vector $\boldsymbol{x}$ (see Section 2.1). The larger BNs were tested with increasing numbers of samples ranging from 25 to 400, while mini-ADAPT was tested with 25 to 2000 samples.

Sharing. Each BN has a different number of parameters that can be shared, where a set of nodes $\boldsymbol{Y}_i \in$

[8]This is a simplification, since our use of the Bernoulli assumes that each CPT is either "correct" or "incorrect." When learned from data, the estimated parameters are clearly almost never exactly correct, but close to or far from their respective original values.

[9]If X is Binomial with parameters n and p, it is well-known that the expected value is $E(X) = np$.

[10]The error probabilities of $T_{\boldsymbol{S}}$ and $T_{\boldsymbol{U}}$ are assumed to be the same as a simplifying assumption.

[11]ADAPT BNs can be found here: `http://works.bepress.com/ole_mengshoel/`.

[12]Our experiments are limited in that we are only learning the parameters of BNs, using data generated from those BNs. Clearly, in most applications, data is not generated from a BN and the true distribution does not conform exactly to some BN structure. However, our analytical and experimental investigation of error would not have been possible without this simplifying assumption.

NAME	$\|\boldsymbol{X}\|$	$\|\boldsymbol{P}\|$	$\|\boldsymbol{W}\|$	CPT
ADAPT_T1	120	26	136	1504
ADAPT_T2	671	107	789	13281
ADAPT_P1	172	33	224	4182
ADAPT_P2	494	99	602	10973
mini-ADAPT	18	3	15	108
Pigs	441	296	592	8427

Table 2: Bayesian networks used in experiments. The $|\boldsymbol{P}|$ column presents the number of potentially shared nodes, with actually shared nodes $\boldsymbol{S} \subseteq \boldsymbol{P}$. The CPT column denotes the total number of parameters in the conditional probability tables.

$\boldsymbol{Y}$ with equal CPTs are deemed sharable. In most cases, there were multiple sets of nodes $\boldsymbol{Y}_1$, ..., $\boldsymbol{Y}_k$ with $|\boldsymbol{Y}_i| \geq 2$ for $k \geq i \geq 1$. When multiple sets were available, the largest set was selected for experimentation, as shown in Section 5.2's pseudo-code.

Metrics. After an EM trial converged to an estimate $\hat{\boldsymbol{\theta}}$, we collected the following three *metrics*:

1. number of iterations t' needed to converge to $\hat{\boldsymbol{\theta}}$,
2. log-likelihood $\ell\ell$ of $\hat{\boldsymbol{\theta}}$, and
3. error: distance between $\hat{\boldsymbol{\theta}}$ and the original $\boldsymbol{\theta}^*$ (see (1) for the definition).

To provide reliable statistics on mean and standard deviation, many randomly initialized EM trials were run.

Software. Among the available software implementations of the EM algorithm for BNs, we have based our work on LibDAI [23].[13] LibDAI uses factor graphs for its internal representation of BNs, and has several BN inference algorithms implemented. During EM, the exact junction tree inference algorithm [16] was used, since it has performed well previously [19].

5.2 VARYING NUMBER OF SHARED NODES

Here we investigate how varying the number of shared nodes impacts EM. A set of hidden nodes $\boldsymbol{H}$ was created for each BN by selecting

$$\boldsymbol{H} = \arg\max_{\boldsymbol{Y}_i \in \boldsymbol{Y}} |\boldsymbol{Y}_i|,$$

where $\boldsymbol{Y}$ is a sharing set partition for BN nodes $\boldsymbol{X}$ (see Section 4.1). In other words, each experimental BN had its largest set of shareable nodes hidden, giving $n_{\boldsymbol{H}} = 12$ nodes for ADAPT_T1, $n_{\boldsymbol{H}} = 66$ nodes for ADAPT_T2, $n_{\boldsymbol{H}} = 32$ nodes for ADAPT_P1, and $n_{\boldsymbol{H}} = 145$ nodes for Pigs.

[13] `www.libdai.org`

The following *gradual sharing method* is used to vary sharing. Given a fixed set of hidden nodes $\boldsymbol{H}$ and an initially empty set of shared nodes $\boldsymbol{S}$:

1. Randomly add $\Delta n_{\boldsymbol{S}} \geq 1$ hidden nodes that are not yet shared to the set of shared nodes $\boldsymbol{S}$. Since we only have a single sharing set, this means moving $\Delta n_{\boldsymbol{S}}$ nodes from the set $\boldsymbol{H} \setminus \boldsymbol{S}$ to the set $\boldsymbol{S}$.
2. Perform m sharing EM trials in this configuration, and record the three metrics for each trial.
3. Repeat until all hidden nodes are shared; that is, $\boldsymbol{S} = \boldsymbol{H}$.

Using the gradual sharing method above, BN nodes were picked as hidden and then gradually shared. When increasing the number of shared nodes, the new set of shared nodes was a superset of the previous set, and a certain number of EM trials was performed for each set.

5.2.1 One Network

For ADAPT_T2, $m = 200$ samples were generated and $n_{\boldsymbol{H}} = 66$ nodes were hidden. We used, in the gradual sharing method, $\Delta n_{\boldsymbol{S}} = 4$ from $n_{\boldsymbol{S}} = 2$ to $n_{\boldsymbol{S}} = 66$ (every hidden node was eventually set as shared).

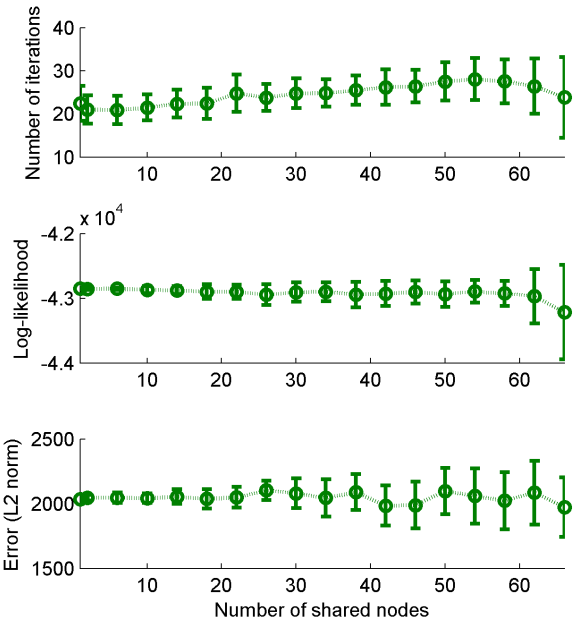


Figure 2: The number of iterations (top), log-likelihood or $\ell\ell$ (middle), and error (bottom) for a varying number of shared nodes $n_{\boldsymbol{S}}$ (along the x-axis) for the BN ADAPT_T2. Here, $n_{\boldsymbol{H}} = 66$ nodes are hidden. Shared nodes are a random subset of the hidden nodes, so $n_{\boldsymbol{S}} \leq n_{\boldsymbol{H}}$.

ADAPT_T1

	Iterations		Likelihood		Error	
Size	$r(\mu)$	$r(\sigma)$	$r(\mu)$	$r(\sigma)$	$r(\mu)$	$r(\sigma)$
25	-0.937	-0.135	0.997	0.882	-0.068	0.888
50	-0.983	-0.765	0.992	0.941	0.383	0.988
100	-0.825	0.702	0.494	0.941	0.786	0.985
200	0.544	0.926	-0.352	0.892	0.517	0.939
400	0.810	0.814	-0.206	0.963	0.863	0.908

ADAPT_T2

	Iterations		Likelihood		Error	
Size	$r(\mu)$	$r(\sigma)$	$r(\mu)$	$r(\sigma)$	$r(\mu)$	$r(\sigma)$
25	0.585	0.852	0.749	0.842	0.276	0.997
50	0.811	0.709	0.377	0.764	0.332	0.981
100	0.772	0.722	-0.465	0.855	-0.387	0.992
200	0.837	0.693	-0.668	0.788	-0.141	0.989
400	0.935	0.677	-0.680	0.784	-0.0951	0.987

ADAPT_P1

	Iterations		Likelihood		Error	
Size	$r(\mu)$	$r(\sigma)$	$r(\mu)$	$r(\sigma)$	$r(\mu)$	$r(\sigma)$
25	-0.769	-0.0818	-0.926	0.278	-0.107	0.774
50	-0.864	-0.0549	-0.939	0.218	-0.331	0.188
100	-0.741	0.436	-0.871	0.678	-0.458	0.457
200	-0.768	0.408	-0.879	0.677	-0.196	0.700
400	-0.665	0.560	-0.862	0.681	0.00444	0.653

PIGS

	Iterations		Likelihood		Error	
Size	$r(\mu)$	$r(\sigma)$	$r(\mu)$	$r(\sigma)$	$r(\mu)$	$r(\sigma)$
25	0.954	0.763	0.970	0.614	0.965	0.887
50	0.966	0.956	0.137	0.908	0.740	0.869
100	-0.922	-0.0167	-0.994	-0.800	-0.893	0.343
200	-0.827	-0.394	-0.985	-0.746	-0.577	0.0157
400	-0.884	-0.680	-0.942	-0.699	-0.339	0.285

Table 3: Values for Pearson's r for the correlation between the number of shared nodes and statistics for these metrics: number of iterations, log-likelihood ($\ell\ell$), and error. $r(\mu)$ defines the correlation between number of shared nodes and the mean of the metric, while $r(\sigma)$ defines the correlation for the standard deviation.

Figure 2 summarizes the results from this experiment. Here, $n_{\boldsymbol{S}}$ is varied along the x-axis while the y-axis shows statistics for different metrics in each of the three sub-plots. For example, in the top plot of Figure 2, each marker is the mean number of iterations (μ), and the error bar is +/- one standard deviation (σ). The main trends[14] in this figure are: parameter sharing increased the mean number of iterations required for EM and slowly decreased the mean $\ell\ell$. Increasing the number of shared nodes resulted in a corresponding increase in standard deviation for the number of iterations, $\ell\ell$, and error of the BN ADAPT_T2. For standard deviation of error, this is in line with our analysis in Section 4.4.

5.2.2 Multiple Networks

We now investigate how varying the number of shared nodes impacts EM for several BNs, specifically the correlation between the number of parameters shared and the mean μ and standard deviation σ for our three metrics. To measure correlation, we use Pearson's sample correlation coefficient r:

$$r(X,Y) = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{(m-1)s_x s_y}, \tag{10}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of two random variables X and Y, and s_x and s_y are the sample standard deviations of X and Y respectively. Here, (10) measures the correlation between m samples from X and Y.[15]

The number of samples, m, refers to the number of (X,Y) sharing samples we have for this correlation analysis (and not the number of samples used to learn BN parameters). In all these experiments, we do a trial for a number of shared nodes, giving several (X,Y) pairs. Consequently, each number of shared nodes tested would be an X, and the metric measured would be a Y. For example, if we use $n_{\boldsymbol{S}} \in \{2,4,6,8,10\}$ shared nodes, then $m = 5$.

We now tie $r(X,Y)$ in (10) to the $r(\mu)$ and $r(\sigma)$ used in Table 3. In Table 3, μ and σ show the mean and standard deviation, respectively, for a metric. Thus, $r(\mu)$ is the correlation of the number of shared nodes and the mean likelihood of a metric, while $r(\sigma)$ is the correlation of the number of shared nodes and the standard deviation of likelihood of a metric.

Figure 2 helps in understanding exactly what is being correlated, as μ and σ for all three metrics are shown for the BN ADAPT_T2. In the top plot, $r(\mu)$ is the correlation between the number of shared nodes (x-axis) and the mean number of iterations (y-axis). In other words, the mean $y_i = \mu_i$ is for a batch of 50 trials of EM. The mean $\bar{y}$ used in Pearson's r is, in this case, a mean of means, namely the mean over 50-EM-trials-means over different numbers of shared nodes.

Table 3 summarizes the experimental results for four BNs. In this table, a positive correlation implies that parameter sharing increased the corresponding metric statistic. For example, the highest correlation between number of shared nodes and mean likelihood is for ADAPT_T1 at 25 samples, where $r(\mu) = 0.997$. This suggests that increasing the number of shared nodes was highly correlated with an increase in the likelihood of EM. Negative coefficients show that increasing the number of shared nodes resulted in a decrease of the corresponding metric statistic.

A prominent trend in Table 3 is the consistently positive correlation between the number of shared nodes

[14] We say "main trends" because the curves for the metrics mean number of iterations and $\ell\ell$ are in fact reversing their respective trends and dropping close to the maximum of 66 shared nodes.

[15] In this case, X is the independent variable, specifically the number of shared nodes. We treat the metric Y as a function of X. When X is highly correlated with Y, this is expressed in r through extreme (positive or negative) correlation values.

NO SHARING						
Observable	Error		Likelihood		Iterations	
	μ	σ	μ	σ	μ	σ
$\boldsymbol{O}_C$	16.325	(-)	-3.047e4	(-)	(-)	(-)
$\boldsymbol{O}_P$	48.38	3.74	-2.009e4	43.94	16.39	4.98
$\boldsymbol{O}_T$	33.25	7.45	-2.492e4	1190	8.65	1.99
SHARING						
Observable	Error		Likelihood		Iterations	
	μ	σ	μ	σ	μ	σ
$\boldsymbol{O}_C$	16.323	(-)	-3.047e4	(-)	(-)	(-)
$\boldsymbol{O}_P$	48.56	3.92	-2.010e4	62.87	15.98	4.95
$\boldsymbol{O}_T$	34.12	14.3	-2.629e4	2630	6.59	2.48

Table 4: Comparison of No Sharing (top) versus Sharing (bottom) for different observable node sets $\boldsymbol{O}_C$, $\boldsymbol{O}_P$, and $\boldsymbol{O}_T$ during 600 EM trials for mini-ADAPT.

$n_{\boldsymbol{S}}$ and the standard deviation of error, $r(\sigma)$, for all 4 BNs. This is in line with the analytical result involving $n_{\boldsymbol{S}}$ in (9).

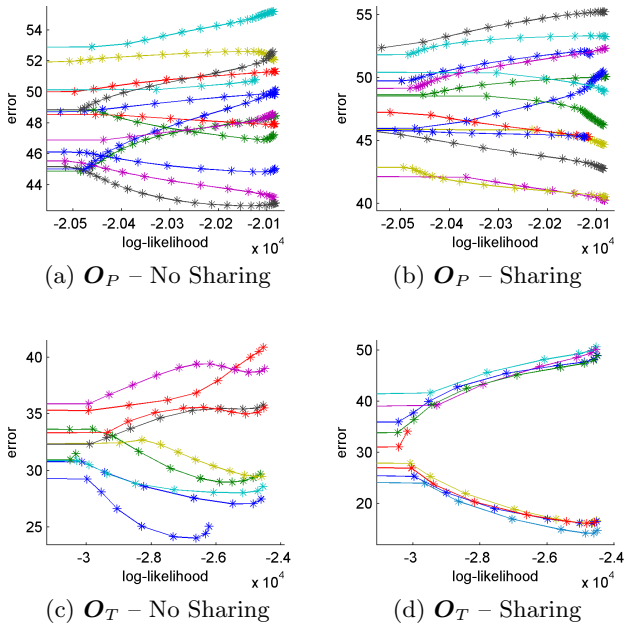
The number of samples was shown to have a significant impact on these correlations. The Pigs network showed a highly correlated increase in the mean number of iterations for 25 and 50 samples. However, for 100, 200, and 400 samples there was a decrease in the mean number of iterations. The opposite behavior is observed in ADAPT_T1, where fewer samples resulted in better performance for parameter sharing (reducing the mean number of iterations), while for 200 and 400 samples we found that parameter sharing increased the mean number of iterations. Further experimentation and analysis may improve the understanding of the interaction between sharing and the number of samples.

5.3 CONVERGENCE REGIONS

5.3.1 Small Bayesian Networks

First, we will show how sharing influences EM parameter interactions for the mini-ADAPT BN shown in Figure 1 and demonstrate how shared parameters jointly converge.

Earlier we introduced $\boldsymbol{O}_P$ as observable nodes in a production system and $\boldsymbol{O}_T$ as observable nodes in a testing system. Complementing $\boldsymbol{O}_P$, hidden nodes are $\boldsymbol{H}_P = \{H_B, H_S, V_B, S_B\}$. Complementing $\boldsymbol{O}_T$, hidden nodes are $\boldsymbol{H}_T = \{V_B, S_B\}$. When a node is hidden, the EM algorithm will converge to one among its potentially many convergence regions. For $\boldsymbol{O}_P$, EM had much less observed data to work with than for $\boldsymbol{O}_T$ (see Figure 1). For $\boldsymbol{O}_P$, the health breaker node H_B was, for instance, not observed or even connected to any nodes that were observed. In contrast, $\boldsymbol{O}_T$ was designed to allow better observation of the components' behaviors, and $H_B \in \boldsymbol{O}_T$. From mini-ADAPT, 500 samples were generated. Depending on the observable set used, either $n_{\boldsymbol{H}} = |\boldsymbol{H}_T| = 2$ or $n_{\boldsymbol{H}} = |\boldsymbol{H}_P| = 4$ nodes were hidden, and 600 random EM trials were executed with and without sharing.



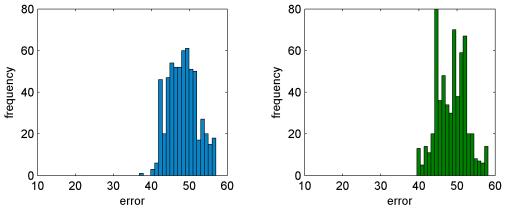
(a) $\boldsymbol{O}_P$ – No Sharing (b) $\boldsymbol{O}_P$ – Sharing

(c) $\boldsymbol{O}_T$ – No Sharing (d) $\boldsymbol{O}_T$ – Sharing

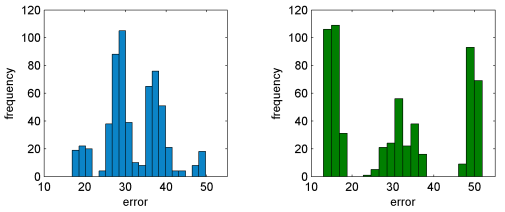
Figure 3: The progress of different random EM trials for the mini-ADAPT BN. Both the degree of sharing (No Sharing versus Sharing) and the number of observables nodes ($\boldsymbol{O}_P$ versus $\boldsymbol{O}_T$) are varied.

Table 4 shows results, in terms of means μ and standard deviations σ, for these EM trials. For $\boldsymbol{O}_P$, with $n_{\boldsymbol{H}} = 4$, the means μ of the metrics error, likelihood, and number of iterations showed minor differences when parameter sharing was introduced. The largest change due to sharing was an increase in σ of likelihood. For $\boldsymbol{O}_T$, where $n_{\boldsymbol{H}} = 2$, differences were greater. The μ of likelihood for sharing was lower with over a 2x increase in σ. The μ for error demonstrated only a minor change, but nearly a 2x increase in σ. This is consistent with our analysis in Section 4.4.

Figure 3a and Figure 3c show how log-likelihood or $\ell\ell$ (x-axis) and error (y-axis) changed during 15 EM trials for $\boldsymbol{O}_P$ and $\boldsymbol{O}_T$ respectively. These EM trials were selected randomly among the trials reported on in Table 4. Parameter sharing is introduced in Figure 3b and Figure 3d. For $\boldsymbol{O}_P$, the progress of the EM trials is similar for sharing (Figure 3b) and non-sharing (Figure 3a), although for a few trials in the sharing condition the error is more extreme (and mostly smaller!). This is also displayed in Table 4, where the difference in number of iterations, error, and likelihood was minor (relative to $\boldsymbol{O}_T$). On the other hand, there is a clear difference in the regions of convergence for $\boldsymbol{O}_T$ when parameter sharing is introduced, consistent with the analysis in Section 4.3. Figure 3d shows how the



(a) $\boldsymbol{O}_P$ – No Sharing (b) $\boldsymbol{O}_P$ – Sharing



(c) $\boldsymbol{O}_T$ – No Sharing (d) $\boldsymbol{O}_T$ – Sharing

Figure 4: Four 20-bin histograms, for mini-ADAPT, of the error values of 600 randomly initialized EM trials at convergence. Both the degree of sharing (No Sharing versus Sharing) and the number of observables nodes ($\boldsymbol{O}_P$ versus $\boldsymbol{O}_T$) are varied.

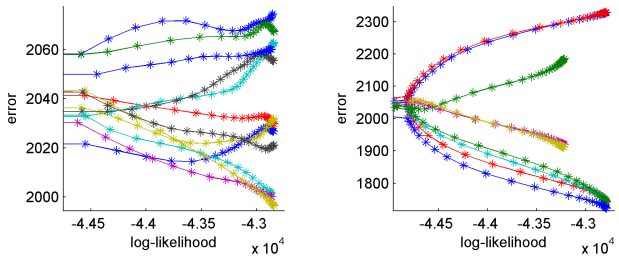
EM trials typically followed a path heading to optima far above or far below the mean error, with two of the EM trials plotted converging in the middle region of the error.

Histograms for the 600 EM trials used in Table 4 are shown in Figure 4. The 20-bin histograms show error $err(\hat{\boldsymbol{\theta}})$ at convergence. The $\boldsymbol{O}_P$ and $\boldsymbol{O}_T$ sets are shown without parameter sharing in Figure 4a and Figure 4c, respectively. Parameter sharing is introduced in Figure 4b and Figure 4d. There is an increased σ of error due to parameter sharing for $\boldsymbol{O}_T$. When comparing Figure 4c (No Sharing) and Figure 4d (Sharing), we notice different regions of error for EM convergence due to sharing. Figure 4c appears to show four main error regions, with the middle two being greatest in frequency, while Figure 4d appears to show three regions of error, with the outer two being most frequent. The outer two regions in Figure 4d are further apart than their non-sharing counterparts, showing that parameter sharing yielded a larger range of error for $\boldsymbol{O}_T$, see Section 4.4.

5.3.2 Large Bayesian Networks

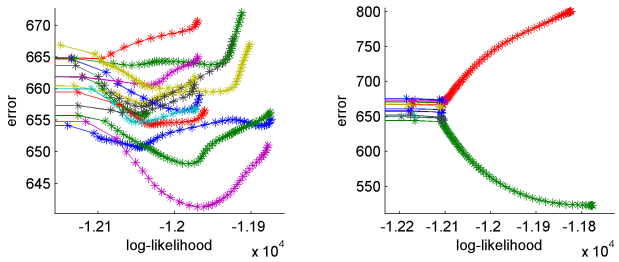
Next, large BNs are used to investigate the effects of parameter sharing, using a varying number of shared nodes. The larger ADAPT networks and Pigs were run with 50 EM trials[16] for each configuration of observable nodes, number of samples, and number of shared nodes. Some of the results are reported here.

[16]The decrease in number of EM trials performed relative to mini-ADAPT was due to the substantial increase in CPU time required (days to weeks).



(a) No Sharing (b) Sharing

Figure 5: The progress of 15 random EM trials for ADAPT_T2. The No Sharing condition (a) shows more locally optimal convergence regions than Sharing (b), where there appears to be only four locally optimal convergence regions.



(a) No Sharing (b) Sharing

Figure 6: The progress of random EM trials for ADAPT_P1. While the number of EM trials is the same for both conditions, the No Sharing condition (a) clearly shows more local optima than Sharing (b).

Figure 5a shows results for ADAPT_T2 without parameter sharing during 15 EM trials using $n_{\boldsymbol{H}} = 66$ hidden nodes and 200 samples. Figure 5b shows a substantial change in error when the $n_{\boldsymbol{H}} = 66$ hidden nodes were shared. The range of the error for EM is much larger in Figure 5b, while the upper and lower error curves have a symmetric quality. Four regions for the converged error are visible in Figure 5b, with the inner two terminating at a lower $\ell\ell$ than the outer two. The lowest error region of Figure 5b is also lower than the lowest error of Figure 5a, while retaining a similar $\ell\ell$.

Figure 6 uses a smaller ADAPT BN, containing 172 nodes instead of 671 nodes (see Table 2). Here, $n_{\boldsymbol{H}} = 33$ nodes were hidden and 200 samples were used. In several respects, the results are similar to those obtained for ADAPT_T2 and mini-ADAPT. However,

Figure 6a shows that EM terminates on different likelihoods, which is not observed in Figure 5a. The error also appears to generally fluctuate more in Figure 6a, whereas the error changes the most during later iterations in Figure 5a. Figure 6b applies parameter sharing to the $n_{\boldsymbol{H}} = 33$ hidden nodes. A symmetric effect is visible between high and low error, reflecting the analysis in Section 4. Of the 15 trials shown in Figure 6b, two attained $\ell\ell > -1.2e^{-4}$, while the rest converged at $\ell\ell \approx -1.21e^{-4}$. Additionally, the $\ell\ell$s of these two trials were greater than any of the non-sharing $\ell\ell$s shown in Figure 6a.

6 CONCLUSION

Bayesian networks have proven themselves as very suitable for electrical power system diagnostics [19,20, 30–33]. By compiling Bayesian networks to arithmetic circuits [4,5], a broad range of discrete and continuous faults can be handled in a computationally efficient and predictable manner. This approach has resulted in award-winning performance on public data from ADAPT, an electrical power system at NASA [30].

The goal of this paper is to investigate the effect, on EM's behavior, of parameter sharing in Bayesian networks. We emphasize electrical power systems as an application, and in particular examine EM for ADAPT Bayesian networks. In these networks, there is considerable opportunity for parameter sharing.

Our results suggest complex interactions between varying degrees of parameter sharing, varying number of hidden nodes, and different dataset sizes when it comes to impact on EM performance, specifically likelihood, error, and the number of iterations required for convergence. One main point, which we investigated both analytically and empirically, is how parameter sharing impacts the error associated with EM's parameter estimates. In particular, we have found analytically that the error variance increases with the number of shared parameters. Experiments with several BNs, mostly for fault diagnosis of electrical power systems, are in line with the analysis. The good news here is that there is, in the sharing case, smaller error some of the time.

Further theoretical research to better understand parameter sharing is required. Since parameter sharing was demonstrated to perform poorly in certain cases, further investigations appear promising. Parameter sharing sometimes reduced the number of EM iterations required for parameter learning, while at other times the number of EM iterations increases. Improving the understanding of the joint impact of parameter sharing and the number of samples on the number of EM iterations would be useful, for example. Finally, it would be interesting to investigate the connection to object-oriented and relational BNs in future work.

References

[1] E.E. Altendorf, A.C. Restificar, and T.G. Dietterich. Learning from sparse data by exploiting monotonicity constraints. In *Proceedings of UAI*, volume 5, 2005.

[2] A. Basak, I. Brinster, X. Ma, and O. J. Mengshoel. Accelerating Bayesian network parameter learning using Hadoop and MapReduce. In *Proc. of BigMine-12*, Beijing, China, August 2012.

[3] P.S. Bradley, U. Fayyad, and C. Reina. Scaling EM (Expectation-Maximization) clustering to large databases. *Microsoft Research Report, MSR-TR-98-35*, 1998.

[4] M. Chavira and A. Darwiche. Compiling Bayesian networks with local structure. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1306–1312, 2005.

[5] A. Darwiche. A differential approach to inference in Bayesian networks. *Journal of the ACM*, 50(3):280–305, 2003.

[6] C.P. de Campos and Q. Ji. Improving Bayesian network parameter learning using constraints. In *Proc. 19th International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, 2008.

[7] B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, (27):94–128, 1999.

[8] 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, Series B*, 39(1):1–38, 1977.

[9] R. Greiner, X. Su, B. Shen, and W. Zhou. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning*, 59(3):297–322, 2005.

[10] S.H. Jacobson and E. Yucesan. Global optimization performance measures for generalized hill climbing algorithms. *Journal of Global Optimization*, 29(2):173–190, 2004.

[11] W. Jank. The EM algorithm, its randomized implementation and global optimization: Some challenges and opportunities for operations research. In F. B. Alt, M. C. Fu, and B. L. Golden, editors, *Perspectives in Operations Research: Papers in Honor of Saul Gass 80th Birthday*. Springer, 2006.

[12] W. B. Knox and O. J. Mengshoel. Diagnosis and reconfiguration using Bayesian networks: An electrical power system case study. In *Proc. of the IJCAI-09 Workshop on Self-⋆ and Autonomous Systems (SAS): Reasoning and Integration Challenges*, pages 67–74, 2009.

[13] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[14] H. Langseth and O. Bangsø. Parameter learning in object-oriented Bayesian networks. *Annals of Mathematics and Artificial Intelligence*, 32(1):221–243, 2001.

[15] S.L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, 19(2):191–201, 1995.

[16] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

[17] W. Liao and Q. Ji. Learning Bayesian network parameters under incomplete data with domain knowledge. *Pattern Recognition*, 42(11):3046–3056, 2009.

[18] G.J. McLachlan and D. Peel. *Finite mixture models*. Wiley, 2000.

[19] O. J. Mengshoel, M. Chavira, K. Cascio, S. Poll, A. Darwiche, and S. Uckun. Probabilistic model-based diagnosis: An electrical power system case study. *IEEE Trans. on Systems, Man, and Cybernetics*, 40(5):874–885, 2010.

[20] O. J. Mengshoel, A. Darwiche, K. Cascio, M. Chavira, S. Poll, and S. Uckun. Diagnosing faults in electrical power systems of spacecraft and aircraft. In *Proceedings of the Twentieth Innovative Applications of Artificial Intelligence Conference (IAAI-08)*, pages 1699–1705, Chicago, IL, 2008.

[21] O. J. Mengshoel, S. Poll, and T. Kurtoglu. Developing large-scale Bayesian networks by composition: Fault diagnosis of electrical power systems in aircraft and spacecraft. In *Proc. of the IJCAI-09 Workshop on Self-⋆ and Autonomous Systems (SAS): Reasoning and Integration Challenges*, pages 59–66, 2009.

[22] O.J. Mengshoel, D.C. Wilkins, and D. Roth. Initialization and restart in stochastic local search: Computing a most probable explanation in Bayesian networks. *IEEE Transactions on Knowledge and Data Engineering*, 23(2):235–247, 2011.

[23] J.M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, August 2010.

[24] S. Natarajan, P. Tadepalli, E. Altendorf, T.G. Dietterich, A. Fern, and A.C. Restificar. Learning first-order probabilistic models with combining rules. In *ICML*, pages 609–616, 2005.

[25] R.S. Niculescu, T.M. Mitchell, and R.B. Rao. Bayesian network learning with parameter constraints. *The Journal of Machine Learning Research*, 7:1357–1383, 2006.

[26] R.S. Niculescu, T.M. Mitchell, and R.B. Rao. A theoretical framework for learning Bayesian networks with parameter inequality constraints. In *Proc. of the 20th International Joint Conference on Artifical Intelligence*, pages 155–160, 2007.

[27] S. Poll, A. Patterson-Hine, J. Camisa, D. Garcia, D. Hall, C. Lee, O.J. Mengshoel, C. Neukom, D. Nishikawa, J. Ossenfort, A. Sweet, S. Yentus, I. Roychoudhury, M. Daigle, G. Biswas, and X. Koutsoukos. Advanced diagnostics and prognostics testbed. In *Proc. of the 18th International Workshop on Principles of Diagnosis (DX-07)*, pages 178–185, 2007.

[28] M. Ramoni and P. Sebastiani. Robust learning with missing data. *Machine Learning*, 45(2):147–170, 2001.

[29] E. Reed and O.J. Mengshoel. Scaling Bayesian network parameter learning with expectation maximization using MapReduce. *Proc. of Big Learning Workshop on Neural Information Processing Systems (NIPS-12)*, 2012.

[30] B. Ricks and O. J. Mengshoel. Diagnosis for uncertain, dynamic and hybrid domains using Bayesian networks and arithmetic circuits. *International Journal of Approximate Reasoning*, 55(5):1207–1234, 2014.

[31] B. W. Ricks, C. Harrison, and O. J. Mengshoel. Integrating probabilistic reasoning and statistical quality control techniques for fault diagnosis in hybrid domains. In *In Proc. of the Annual Conference of the Prognostics and Health Management Society 2011 (PHM-11)*, Montreal, Canada, 2011.

[32] B. W. Ricks and O. J. Mengshoel. Methods for probabilistic fault diagnosis: An electrical power system case study. In *Proc. of Annual Conference of the PHM Society, 2009 (PHM-09)*, San Diego, CA, 2009.

[33] B. W. Ricks and O. J. Mengshoel. Diagnosing intermittent and persistent faults using static Bayesian networks. In *Proc. of the 21st International Workshop on Principles of Diagnosis (DX-10)*, Portland, OR, 2010.

[34] A. Saluja, P.K. Sundararajan, and O.J. Mengshoel. Age-Layered Expectation Maximization for parameter learning in Bayesian Networks. In *Proceedings of Artificial Intelligence and Statistics (AIStats)*, La Palma, Canary Islands, 2012.

[35] B. Thiesson, C. Meek, and D. Heckerman. Accelerating EM for large databases. *Machine Learning*, 45(3):279–299, 2001.

[36] S. Watanabe. Algebraic analysis for nonidentifiable learning machines. *Neural Computation*, 13(4):899–933, 2001.

[37] C.F. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[38] Z. Zhang, B.T. Dai, and A.K.H. Tung. Estimating local optimums in EM algorithm over Gaussian mixture model. In *Proc. of the 25th international conference on Machine learning*, pages 1240–1247. ACM, 2008.