Learning Bayesian Networks using the Constrained Maximum a Posteriori Probability Method

Yu Yang, Xiaoguang Gao, Zhigao Guo, Daqing Chen

PII: S0031-3203(19)30070-6
DOI: https://doi.org/10.1016/j.patcog.2019.02.006
Reference: PR 6812

To appear in: *Pattern Recognition*

Received date: 24 May 2018
Revised date: 2 February 2019
Accepted date: 7 February 2019

Please cite this article as: Yu Yang, Xiaoguang Gao, Zhigao Guo, Daqing Chen, Learning Bayesian Networks using the Constrained Maximum a Posteriori Probability Method, *Pattern Recognition* (2019), doi: https://doi.org/10.1016/j.patcog.2019.02.006

Highlights

- This paper proposed a frame work based on the inequality constrained optimization model to learn conditional probability table parameters by incorporating expert judgments and Dirichlet priors.
- We further improve the proposed method by developing a constrained Bayesian Dirichlet prior.
- Combined the proposed method, we provide an improved expectation maximum algorithm for learning conditional probability table parameters from incomplete data.
- The contributed algorithm is tested on 13 well-known Bayesian networks, whose parameter number varies from 9 to 1157. The experiments show that the proposed method outperforms most of the existing parameter learning algorithms, especially when training data are extremely scarce.
- A real facial action unit recognition case with incomplete data is conducted. The results show that the proposed method can build a more accurate Bayesian network for recognizing facial action units.

Learning Bayesian Networks using the Constrained Maximum a Posteriori Probability Method

Yu Yang[a], Xiaoguang Gao[a,*], Zhigao Guo[a], Daqing Chen[b]

[a] *School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China*
[b] *School of Engineering, London South Bank University, London, UK*

Abstract

Purely data-driven methods often fail to learn accurate conditional probability table (CPT) parameters of discrete Bayesian networks (BNs) when training data are scarce or incomplete. A practical and efficient means of overcoming this problem is to introduce qualitative parameter constraints derived from expert judgments. To exploit such knowledge, in this paper, we provide a constrained maximum a posteriori (CMAP) method to learn CPT parameters by incorporating convex constraints. To further improve the CMAP method, we present a type of constrained Bayesian Dirichlet priors that is compatible with the given constraints. Combined with the CMAP method, we propose an improved expectation maximum algorithm to process incomplete data. Experiments are conducted on learning standard BNs from complete and incomplete data. The results show that the proposed method outperforms existing methods, especially when data are extremely limited or incomplete. This finding suggests the potential effective application of CMAP to real-world problems. Moreover, a real facial action unit (AU) recognition case with incomplete data is conducted by applying different parameter learning methods. The results show that the recognition accuracy of respective recognition methods can be improved by the AU BN, which is trained by the proposed method.

*Corresponding author
Email addresses: youngiv@126.com (Yu Yang), cxg2012@nwpu.edu.cn (Xiaoguang Gao), buckleyguo@mail.nwpu.edu.cn (Zhigao Guo), chend@lsbu.ac.uk (Daqing Chen)

Keywords: Bayesian network, parameter learning, expert judgment, facial action unit

1. Introduction

Bayesian networks (BN) [1] have become an efficient tool to express and infer uncertain knowledge. A discrete BN consists of a directed acyclic graph (DAG) and a set of related conditional probability table (CPT) parameters. The DAG qualitatively expresses (in)dependency relationships among variables, while the CPTs probabilistically quantify those relationships.

The first step in building a BN from data (or samples) is to recover the DAG. When training data are scarce or incomplete, it is unrealistic to build a DAG by purely data-driven algorithms [2–5]. To address this problem, some systematic approaches have been established to help domain experts artificially define BN structures (DAGs) [2, 6]. However, few experts have the confidence to directly provide CPTs for a BN as the corresponding structure identified in advance. Moreover, a scarce or incomplete dataset alone is insufficient for accurately revealing the CPTs relating to a known structure.

In addition to structures, domain experts might be able to provide qualitative judgments about parameters [7]. It has been proved that expert judgments are helpful for improving parameter learning accuracy when data are scarce [8–13]. In practice, qualitative constraints can be derived from expert judgments. These constraints are almost linear, and are thus convex. Although concave constraints exist, such as $\theta_{ijk} \neq 0.5$, experts actually have a small probability of providing judgments that will derive such constraints. Thus, in this paper, we only consider convex constraints, while emphasizing linear constraints.

Accordingly, we concentrate on enhancing the accuracy of learning CPT parameters from scarce or incomplete data by incorporating convex parameter constraints as structures that have been artificially defined.

Several methods have been applied to learn CPT parameters from scarce data by incorporating qualitative constraints, including those listed below.

- Convex Optimization (CO) [9, 10, 14–19]: This method is an extension of the maximum likelihood (ML); however, CO partly alleviates the overfitting problem by introducing constraints. There are two approaches to assimilate the information of constraints: (a) penalty functions constructed from constraints are used to modify likelihood functions [9, 14]; (b) constraints are directly used to restrict parameter spaces [10, 19]. Theoretically, the CO method can cope with all convex constraints.

- Isotonic Regression (IR) [11, 20]: IR computes isotonic estimations by the minimum lower sets (MLS) [21] algorithm based on data statistics and monotonic influences. Then it takes the isotonic estimations as desired CPT parameters.

- Qualitative Maximum a Posteriori (QMAP) [12]: This method firstly recruits Monte Carlo samples from the constrained parameter space to construct prior Dirichlet priors. Next, it respectively copes with them by using the maximum a posteriori (MAP) algorithm to obtain the MAP estimations. It finally takes the mean value of the MAP estimations as the learned BN parameters.

- Multinomial Parameter Learning with Constraints (MPL-C) [13, 22, 23]: MPL-C was recently proposed to learn CPTs by creatively reconstructing auxiliary BNs, which are hybrid BNs [24], to infer the posterior distribution of BN parameters. It then takes the expectation as the parameter estimation.

In this paper, we propose a framework—a constrained maximum a posteriori (CMAP) method—to address CPT learning by incorporating convex constraints and Dirichlet priors. CMAP is proposed based on a convex optimization method, in which given constraints are directly used to restrict feasible parameter spaces. Dirichlet priors are introduced to further alleviate the overfitting problem of the basic convex optimization method. Although BDeu prior [25] and flat prior [9] are commonly used prior distributions for discrete variables, they are often incompatible with expert judgments as they always drive conditional distributions moving towards uniforms. Consequently, they likely restrict parameter learning accuracy.

Accordingly we develop constrained Bayesian Dirichlet (CBD) priors that are compatible with expert judgments. The convex optimization problem is approximately solved by a barrier method with a guaranteed specified accuracy.

A scenario where some records are missing or nodes are unobservable leads to incomplete data. If the missing records randomly occur and there remains adequate complete data, we can remove incomplete samples to obtain a complete dataset. At that point, a complete-data-driven algorithm can be used to learn CPTs. Otherwise, we can employ the classic expectation maximization (EM) algorithm [26] to learn the CPT parameters from such incomplete data. However, the learning results by using the EM algorithm are often frustrating. When a dataset is incomplete, the expectation of the likelihood function is actually multimodal. The EM algorithm is essentially a special hill-climbing method that can be applied to such parameter learning tasks. Thus, a local optimum is always found. The local optimum parameter cannot guarantee that a 'good' BN is built since the global optimal solution is even not a desired result. Furthermore, because the same dataset probably results in contradictory BNs with the change of the initial condition (or start point), it is unreliable to parameterize a DAG from incomplete data by a purely data-driven method.

It has been shown that expert judgments are additionally helpful to improve the accuracy of learning parameters from incomplete data [10, 14]. The parameter constraints can restrict the path by which the EM procedures converge to a local optimum. Thus, even though different BNs will be learned with the start point changing, each of them can satisfied experts' preferences. Similar to the ML algorithm, the proposed approach is compatible with the EM algorithm since the convergence can be guaranteed.

The remainder of this paper is structured as follows. Section 2 outlines basic information on BNs. Section 3 lists linear constraints that can be collected from expert judgments. Section 4 describes our framework for learning CPTs from both complete and incomplete data. Section 5 compares different methods by learning 13 standard BNs and training a real facial action unit (AU) recognition model. Section 6 concludes this paper.

2. Preliminaries

A discrete Bayesian network consists of a DAG G and related CPT parameters θ. The $G = (X, E)$ expresses independence relationships among a set of variables (or nodes) $X = \{X_1, X_2, \cdots, X_n\}$, where $E = \{X_j \to X_i | X_j \in \Pi_i,\ i = 1, \cdots, n\}$ is the set of arrows in the DAG and $\Pi_i \subseteq X \setminus \{X_i\}$ is the parent set for X_i. In other words, there exists an arrow in G points to X_i from each node in Π_i. The $\theta = \{\theta_1, \theta_2, \cdots, \theta_n\}$ quantifies the dependency relationships among X, where θ_i is the CPT related to family G_i. According to the Markov independency, given all parents, X_i is independent of its other non-descendant nodes. Thus, we have

$$p(X) = \prod_{i=1}^{n} p(X_i | \Pi_i) \tag{1}$$

As a result, the goal of parameter learning on BNs is to determine each conditional distribution $p(X_i|\Pi_i)$. For simplicity, we define θ_{ijk} as a specific conditional probability $p(X_i = k|\Pi_i = j)$, where $k \in \{1, 2, \cdots, r_i\}$ is the state of node X_i, and $j \in \{1, 2, \cdots, q_i\}$ expresses the configuration of the parent set Π_i. Thus, parameters of X_i construct a $r_i \times q_i$ conditional probability table (CPT). Given data $D = \{D_l | l = 1, 2, \cdots, N\}$, the log-likelihood function for θ is

$$L(\theta, D) = \log p(D|\theta) = \sum_{i=1}^{N} \log p(D_l|\theta) \tag{2}$$

We respectively define N_{ijk} as the count for records where and $X_i = k$ and $\Pi_i = j$, and $N_{ij} = \sum_k N_{ijk}$ as the count for records where $\Pi_i = j$ in D. Hence, the maximum likelihood estimation (MLE) is obtained by maximizing $L(\theta, D)$:

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}} \tag{3}$$

Obviously, the ML method will fail to work if $N_{ij} = 0$ (i.e., the certain parent configuration $\Pi_i = j$ has not appeared in data D). In practice, it is common that some parent

configurations will scarcely appear, even for a large number of data [22]. Then a Dirichlet prior is introduced to overcome this problem:

$$p(\theta|G) \propto \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}\theta_{ijk}^{\tau_{ijk}-1} \tag{4}$$

where $\tau = \{\tau_{ijk}\}$ is the hyper-parameter set and $\tau_{ij} = \sum_k \tau_{ijk}$. Flat prior ($\tau_{ijk} - 1 = 1$) or BDeu prior ($\tau_{ijk} - 1 = \frac{1}{r_i q_i}$) are popular Dirichlet priors [9]. Then the objective function can be a logarithmic form of the conditional distribution of θ given D:

$$\log p(\theta|D) = \log p(D|\theta)p(\theta) + c \tag{5}$$

Here, c is a constant. Therefore, we can obtain the MAP estimation for a single parameter by maximizing $\log p(\theta|G, D)$:

$$\hat{\theta}_{ijk} = \frac{N_{ijk} + \tau_{ijk} - 1}{N_{ij} + \tau_{ij} - r_i} \tag{6}$$

3. Linear Parameter Constraints

Example 1 *From the widely accepted judgment that "people who smoke have a higher risk of developing lung cancer than those who do not", we can obtain a parameter constraint as*

$$p(Cancer = true|Smoke = false) \leq p(Cancer = true|Smoke = true).$$

Like Example 1, an expert judgment can induce qualitative parameter constraints. An *interior constraint* restricts the parameters that share the same parent state configuration within a CPT column, such as $\theta_{ijk} \leq 0.1$ and $\theta_{ijk_1} \leq \theta_{ijk_2}$. Nevertheless, an *exterior constraint* expresses an inequality relationship across two or more CPT columns, which means the constrained parameters have different parent configurations, such as $\theta_{ij_1k_1} \leq \theta_{ij_2k_2}$. Both interior and exterior constraints have been proved to be helpful for improving

parameter learning accuracy, especially when training data are limited or incomplete [9].

Regardless of their interior or exterior status, the parameter constraints derived from practical expert judgments can almost be formulated as a linear inequality

$$f(\theta) \leq 0 \tag{7}$$

where $f : \mathbb{R}^{dim(BN)} \rightarrow \mathbb{R}$ is a linear function and $dim(BN)$ is the number of free parameters of BN. Several specific types of constraints can be derived from the linear inequality.

Range Constraints A range constraint defines the upper or lower bound (or both) of a single parameter, which can be represented as

$$0 \leq \alpha \leq \theta_{ijk} \leq \beta \leq 1 \tag{8}$$

Inequality Constraints An inequality constraint defines the relative relation between a pair of parameters, that is

$$\theta_{i'j'k'} \leq \alpha\theta_{ijk} + \beta \tag{9}$$

where two groups of subscripts must be different and $0 \leq \alpha\theta_{ijk} + \beta \leq 1$.

Additive Inequality Constraints An additive inequality constraint is the relative relation between two summations of parameters. A qualitative influence gives a typical additive inequality constraint as

$$\sum_{k=1}^{k^{(c)}} \theta_{isk} \leq \sum_{k=1}^{k^{(c)}} \theta_{itk} \tag{10}$$

where $k^{(c)} \in \{1, 2, \cdots, r_i\}$. Moreover, s and t express two parent configurations, where only the state of the concerned parent in the qualitative influence relationship changes but the configurations for the other parents remain the same.

Example 2 *A simple BN* $X_1 \xrightarrow{+} X_2 \xleftarrow{-} X_3$ *includes a positive influence and a negative influence. Thus, for* $k^{(c)} \in \{1, 2, \cdots, r_2\}$*, we have two groups of constraints as*

$$X_1 \xrightarrow{+} X_2 : \sum_{k=1}^{k^{(c)}} p(X_2 = k | X_1 = 1, X_3) \leq \sum_{k=1}^{k^{(c)}} p(X_2 = k | X_1 = 2, X_3) \leq \cdots$$
$$X_3 \xrightarrow{-} X_2 : \sum_{k=1}^{k^{(c)}} p(X_2 = k | X_1, X_3 = 1) \geq \sum_{k=1}^{k^{(c)}} p(X_2 = k | X_1, X_3 = 2) \geq \cdots$$

Axiomatic Constraints Probabilities should be normalized and nonnegative:

$$\begin{cases} \sum_{k=1}^{r_i} \theta_{ijk} = 1 \\ 0 \leq \theta_{ijk} \leq 1 \end{cases} \tag{11}$$

4. Constrained Maximum a Posteriori Method

4.1. Learning from Complete Data

4.1.1. Learning CPTs by Using Constraints and Dirichlet Priors

It can be seen that $f(x) = n \log x$ $(n \geq 0,\ x > 0)$ is a concave function, since the derivative $f'(x) = \frac{n}{x}$ decreases as x increases. Hence, the log-likelihood function is a concave (a positive sum of concave functions is also concave). Then parameter learning can be modeled as a standard convex optimization problem if the feasible parameter space is convex [27–29]. For constrained maximum likelihood (CML) [10] model, constraint set $\Omega = \{f_l(\theta) \leq 0 | l = 1, \cdots, m\}$ is directly taken as optimization constraints. Thus, we have

$$\begin{aligned} &\arg\min_{\theta} \ -\log p(D|\theta) \\ &s.t. \ \sum_{k=1}^{r_i} \theta_{ijk} = 1,\ i = 1, \cdots, n; j = 1, \cdots, q_i \\ &\quad 0 \leq \theta_{ijk} \leq 1,\ i = 1, \cdots, n; j = 1, \cdots, q_i; k = 1, \cdots, r_i \\ &\quad f_l(\theta) \leq 0,\ l = 1, \cdots, m \end{aligned} \tag{12}$$

When some parent configurations are absent from D, the problem becomes a combination of a feasibility problem (wherein the parent configurations are absent) [30] and a convex optimization problem (wherein the parent configurations are present). This problem can still be solved by some convex optimization techniques; nevertheless, the solution for the feasibility problem is probably undesired. A Dirichlet prior can hence be introduced to mediate this problem. Accordingly,

$$\log p(\theta|D) = \log p(D|\theta)p(\theta) - \log p(D) \tag{13}$$

Removing the constant term $-\log p(D)$, the objective function can be substituted as

$$\begin{aligned} &\arg\min_{\theta} \ -\log p(D|\theta)p(\theta) \\ &s.t. \ \sum_{k=1}^{r_i} \theta_{ijk} = 1, \ i = 1, \cdots, n; j = 1, \cdots, q_i \\ &\quad 0 \leq \theta_{ijk} \leq 1, \ i = 1, \cdots, n; j = 1, \cdots, q_i; k = 1, \cdots, r_i \\ &\quad f_l(\theta) \leq 0, \ l = 1, \cdots, m \end{aligned} \tag{14}$$

Widely-used flat and BDeu priors are often incompatible with constraints as they always drive local conditional distributions to move towards uniforms. Then the learning accuracy or prediction accuracy of learned BNs may be impeded. To further improve learning accuracy, we introduce constrained Bayesian Dirichlet (CBD) priors from constraints as

$$p(\theta|G) \propto \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\gamma_{ij}\overline{\theta}_{ijk}} \tag{15}$$

where γ_{ij} is the weight of the prior and $\overline{\theta} = \{\overline{\theta}_{ijk}\}$ is the mean value of the constrained parameter space. Thus, $\overline{\theta}$ can be computed as

$$\overline{\theta} = \frac{\int_{\Omega} \theta d\theta}{\int_{\Omega} d\theta} \tag{16}$$

The integration is sometimes difficult to compute directly. If it is practical to generate an adequate number of samples from constrained parameter space, we can employ a Monte Carlo method to approximate the integral [31]. Otherwise, a strictly feasible point can be a relatively accurate approximation for $\overline{\theta}$ since the constrained space is very narrow.

We refer to the proposed approach as the constrained maximum a posteriori (CMAP) method. The global optimal solution for equation (14) can be found in polynomial time in the input size [10, 32] by Newton's method. In addition, non-linear convex constraints are theoretically allowed as long as they are second-order differentiable.

4.1.2. Solving by Newton's Method

To solve equation (14) with the classical *Newton's method* [27], we first transform it into an unconstrained problem. Letting $f_0(\theta) = -\log p(D|G,\theta)p(\theta|G)$ and $\theta_{ijr_i} = 1 - \sum_{k=1}^{r_i-1}\theta_{ijk}$, we make the inequality constraints implicit in the objective:

$$\arg\min_{\theta} \; f_0(\theta) + \sum_{l=1}^{m} I_-(f_l(\theta)) \tag{17}$$

where the constraint $0 \le \theta_{ijk} \le 1$ and $\sum_{k=1}^{r_i-1}\theta_{ijk} \le 1$ is implicated, and $I_- : \mathbb{R} \to \mathbb{R}$ is an indicator function for the non-positive real numbers:

$$I_-(x) = \begin{cases} 0, & x \le 0 \\ +\infty, & x > 0 \end{cases} \tag{18}$$

Although equation (17) has no constraints, its objective function is not second-order differentiable. Thus, Newton's method cannot be directly applied. To cope with this problem, we can approximate the indicator function I_- by a logarithmic barrier function

$$\hat{I}_-(x) = -\frac{1}{u}\log(-x), \quad u > 0 \tag{19}$$

Here, u is a factor that is used to control the approximation accuracy. As u increases, the

approximation becomes more accurate. Similar to I_-, $\hat{I}_-$ is strictly convex. Furthermore, $\hat{I}_-$ has a more attractive property such that it is continuous second-order differentiable. Substituting $\hat{I}_-$ for I_- in equation (17), we obtain the approximation

$$\arg\min_{\theta} \ f_0(\theta) + \sum_{l=1}^{m} \frac{-1}{u} \log(-f_l(\theta)) \tag{20}$$

For convenience, we define $J(\theta)$ as the objective in equation (20). As $J(\theta)$ is convex and second-order differentiable, the positive definiteness of Hessian matrix $\nabla^2 J(\theta)$ implies

$$-\nabla J(\theta)^{\mathrm{T}} \nabla^2 J(\theta)^{-1} \nabla J(\theta) \leq 0 \tag{21}$$

where $\nabla J(\theta)^{\mathrm{T}}$ is the transposed matrix of $\nabla J(\theta)$, $\nabla^2 J(\theta)^{-1}$ denotes the inverse matrix of $\nabla^2 J(\theta)$, and the equality holds if and only if $\nabla J(\theta) = 0$. Thus, the *Newton step* $\triangle\theta_{\mathrm{nt}} = -\nabla^2 J(\theta)^{-1} \nabla J(\theta)$ is a descent direction. It reveals that we can find the optimal value for θ along $\triangle\theta_{\mathrm{nt}}$ from a strictly feasible point.

Letting $\tilde{N}_{ijk} = N_{ijk} + \tau_{ijk} - 1$, the partial derivative of $J(\theta)$ with respect to θ_{ijk} is

$$\frac{\partial J(\theta)}{\partial \theta_{ijk}} = \frac{\tilde{N}_{ijr_i}}{\theta_{ijr_i}} - \frac{\tilde{N}_{ijk}}{\theta_{ijk}} - \frac{1}{u} \sum_{l=1}^{m} \frac{1}{f_l(\theta)} \frac{\partial f_l(\theta)}{\partial \theta_{ijk}} \tag{22}$$

Therefore, $\nabla J(\theta) = (\frac{\partial J(\theta)}{\partial \theta_{ijk}})$ is a $dim(BN)$ column vector, where $dim(BN)$ is the number of free parameters of the BN. The second-order partial derivative of $J(\theta)$ with respect to θ_{ijk} and $\theta_{i'j'k'}$ can be expressed as

$$\frac{\partial^2 J(\theta)}{\partial \theta_{ijk} \partial \theta_{i'j'k'}} = \begin{cases} \dfrac{\tilde{N}_{ijr_i}}{\theta_{ijr_i}^2} + \dfrac{\tilde{N}_{ijk}}{\theta_{ijk}^2} + \dfrac{1}{u} \displaystyle\sum_{l=1}^{m} (\frac{1}{f_l^2(\theta)} \frac{\partial f_l(\theta)}{\partial \theta_{ijk}} - \frac{1}{f_l(\theta)} \frac{\partial^2 f_l(\theta)}{\partial \theta_{ijk}^2}), & ijk = i'j'k' \\ \dfrac{\tilde{N}_{ijr_i}}{\theta_{ijr_i}^2} + \dfrac{1}{u} \displaystyle\sum_{l=1}^{m} (\frac{1}{f_l^2(\theta)} \frac{\partial f_l(\theta)}{\partial \theta_{ijk}} - \frac{1}{f_l(\theta)} \frac{\partial^2 f_l(\theta)}{\partial \theta_{ijk}^2}), & ij = i'j' \wedge k \neq k' \\ \dfrac{1}{u} \displaystyle\sum_{l=1}^{m} (\frac{1}{f_l^2(\theta)} \frac{\partial f_l(\theta)}{\partial \theta_{i'j'k'}} - \frac{1}{f_l(\theta)} \frac{\partial^2 f_l(\theta)}{\partial \theta_{ijk} \partial \theta_{i'j'k'}}), & ij \neq i'j' \end{cases} \tag{23}$$

Thus, $\nabla^2(\theta) = (\frac{\partial^2(\theta)}{\partial\theta_{ijk}\partial\theta_{i'j'k'}})$ is a $dim(BN) \times dim(BN)$ matrix. If we only gather linear constraints, the second-order derivative of $f_l(\theta)$ is always zero. Then $\frac{\partial^2(\theta)}{\partial\theta_{ijk}\partial\theta_{i'j'k'}}$ can be furthermore simplified as

$$\frac{\partial^2 J(\theta)}{\partial\theta_{ijk}\partial\theta_{i'j'k'}} = \begin{cases} \frac{\tilde{N}_{ijr_i}}{\theta^2_{ijr_i}} + \frac{\tilde{N}_{ijk}}{\theta^2_{ijk}} + \frac{1}{u}\sum_{l=1}^{m}\frac{1}{f_l^2(\theta)}\frac{\partial f_l(\theta)}{\partial\theta_{ijk}}, & ijk = i'j'k' \\ \frac{\tilde{N}_{ijr_i}}{\theta^2_{ijr_i}} + \frac{1}{u}\sum_{l=1}^{m}\frac{1}{f_l^2(\theta)}, & ij = i'j' \wedge k \neq k' \\ \frac{1}{u}\sum_{l=1}^{m}\frac{1}{f_l^2(\theta)}\frac{\partial f_l(\theta)}{\partial\theta_{i'j'k'}}, & ijk \neq i'j'k' \end{cases} \tag{24}$$

At this point, based on the gradient discussed in equations (22–24), the approximate problem shown in equation (20) can be solved by an improved Newton's method, the barrier method [27]. Defining $\theta^*(u)$ as the solution of equation (20) and f^* as the lower bound on the optimal value, the accuracy of the approximation is given by [27]

$$f_0(\theta^*(u)) - f^* \leq \frac{m}{u} \tag{25}$$

where m is the number of constraints.

4.2. Learning from Incomplete Data

An incomplete dataset means some of the records are missing, or some nodes are unobservable (or hidden). If the dataset is $D = \{D_l | l = 1, \cdots, N\}$, then D_l is the l-th sample. Respectively, define $D_l^{(m)} = \{D_{l,1}^{(m)}, \cdots, D_{l,u}^{(m)}\}$ and $D_l^{(o)} = \{D_{l,1}^{(o)}, \cdots, D_{l,v}^{(o)}\}$ as the missing part and observed part of D_l, then we have $D_l^{(m)} \cap D_l^{(o)} = \varnothing$ and $D_l^{(m)} \cup D_l^{(o)} = D_l$. As $D_l^{(m)}$ $(l = 1, \cdots, N)$ is unknown, the expectation of likelihood function $\log p(D|\theta)$ becomes a combination of those likelihood functions that are related to all possible instantiations of incomplete data D. Although a likelihood function for complete data is unimodal, the expectation of $\log p(D|\theta)$ is multimodal since different instantiations of D make the mode of $\log p(D|\theta)$ various. Thus, the related optimization problem is not a convex model [33].

If missing records randomly occur and there remain an adequate number of complete samples, we can remove incomplete samples to obtain a complete dataset. That is, for $l = 1, \cdots, N$, if $D_l^{(m)} \neq \varnothing$, $D = D \setminus D_l$. Then, complete-data-driven algorithms can be used to learn CPTs. However, when records are not randomly missing or data are scarce, this approach becomes impractical. The EM algorithm is a conventional technique to learn parameters from incomplete data [26, 33], which can iteratively reach a local maximum of the expectation of $\log p(D|\theta)$. The standard EM algorithm is comprised of two key steps:

- *E-step*: Compute the expectation of the log-likelihood function based on the incomplete data D and current parameter estimation $\theta^{(t)}$, which is updated in the M-step:

$$Q(\theta|\theta^{(t)}) = E_{\theta^{(t)}}[\log p(D|\theta)|\theta^{(t)}, D]$$

- *M-step*: Maximize the current expectation $Q(\theta|\theta^{(t)})$, which is updated in the E-step, to determine the new parameter:

$$\theta^{(t+1)} = \arg\min_{\theta} -Q(\theta|\theta^{(t)})$$

The EM algorithm can start from either the E-step when an initial parameter θ^0 is defined, or from the M-step by artificially assigning missing records.

However, the EM algorithm is often trapped in undesired local optimal solutions, and different start points may result in contradictory BNs. Parameter constraints and Dirichlet priors are helpful for deriving an EM procedure converging to the local optimum where the learned BN satisfies domain knowledge. Given a Dirichlet prior $p(\theta)$, the expectation can be modified as $Q(\theta|\theta^{(t)}) = E_{\theta^{(t)}}[\log p(\theta|D)|\theta^{(t)}, D]$. According to equation (16), we have

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= E_{\theta^{(t)}}[\log p(D|\theta) + \log p(\theta) - \log p(D)|\theta^{(t)}, D] \\ &= E_{\theta^{(t)}}[\log p(D|\theta)|\theta^{(t)}, D] + \log p(\theta) - \log p(D) \end{aligned}$$

Let

$$Q'(\theta|\theta^{(t)}) = E_{\theta^{(t)}}[\log p(D|\theta)|\theta^{(t)}, D] + \log p(\theta) \tag{26}$$

It has the same optimization as $Q(\theta|\theta^{(t)})$ for a shared feasible domain. As $Q'(\theta|\theta^{(t)})$ is concave, given convex parameter constraints, the improved M-step can still obtain the global optimum solution. Then, the modified EM algorithm shares convergence and optimality properties (more details in Appendix A). However, the EM procedure likely stops at a point where the gradient is orthogonal to the constraints instead of a stationary point with a zero gradient [33]. The improved EM algorithm is shown in Algorithm 1.

Input: incomplete data D, constraints Ω, initial parameter $\theta^{(0)}$, tolerance $\epsilon > 0$
Output: parameter estimation θ

Let $t = 0$;
repeat
  E-step:
    Compute $Q'(\theta|\theta^{(t)})$ based on equation (26).
  M-step:
    Compute $\theta^{(t+1)}$ by calling Algorithm 2 to solve problem $\theta^{(t+1)} = \arg\min_\theta \ -Q'(\theta|\theta^{(t)})$ subject to Ω.
  Increase t: $t = t + 1$.
until $Q'(\theta^{(t+1)}|\theta^{(t)}) - Q'(\theta^{(t)}|\theta^{(t-1)}) < \epsilon$;
return $\theta^{(t)}$

Algorithm 1: Improved EM algorithm

5. Experimental Evaluation

The conducted experiments consist of a group of standard BN learning cases and a real-world case study. The standard BN learning cases compare different parameter learning methods by measuring the errors from known true CPTs to the CPTs learned from complete scarce data. The comparison can show the potentials of those methods being applied to real-world problems. However, there are no true CPTs in the real case study.

Thus, the prediction performance (AUC) metric is employed to evaluate BN models built from the same incomplete data by different parameter learning algorithms.

To show the performance of the proposed method, we consider the following methods:

- Conventional parameter learning algorithms: ML (equation 3) and MAP (equation 6, using the flat prior)
- Constrained maximum likelihood (CML) algorithm (equation 12)
- Proposed algorithm: CMAP (equation 14, using the flat prior) and CMAP+ (equation 14, using the CBD prior with $\gamma_{ij} = r_i$)

5.1. Experiments on Standard BNs

5.1.1. Complete Data

The proposed method was compared with other methods by learning 13 standard BNs in this group of experiments. Except the Boerlage92 BN [34], the other standard BNs are publicly available in the BN repository[1]. They range from typically small expert-built BNs to those that are as large as what could be reasonably produced by experts.

The *Kullback-Leibler (K-L) divergence* metric [35] was selected as the criterion for evaluating errors between true CPTs and estimated CPTs. To avoid $\log 0$ in the computation of K-L divergences, zero values in CPTs were replaced by a tiny value (1×10^{-10}). For a BN, averaged K-L divergence is used, which is computed as

$$\overline{KL}(\theta, \hat{\theta}) = \frac{1}{\sum_{i=1}^{n} r_i q_i} \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \theta_{ijk} \log(\frac{\theta_{ijk}}{\hat{\theta}_{ijk}}) \tag{27}$$

The experiment settings are summarized follows:

- For the standard BNs, the structures and true CPTs were known but there were

[1] http://www.bnlearn.com/bnrepository/

no true data. Thus, training datasets were randomly sampled from true CPTs with different sparsity levels (50, 100, and 500).

- Parameter constraints on standard BNs were synthesized according to the true CPTs as constraint definitions are satisfied [8]. For a parameter θ_{ijk} whose true value is greater than 0.9, a range constraint was generated as $0.9 \leq \theta_{ijk} \leq 1$. For two parameters θ_{ijk} and $\theta_{ij'k'}$ from the same CPT row or column, if their true values satisfy $\theta_{ijk} - \theta_{ij'k'} \leq -0.2$, an inequality constraint was generated as $\theta_{ijk} \leq \theta_{ij'k'}$. Moreover, the maximum number of constraints for a CPT was 20.

- Learning was repeated 20 times for each BN and data size, and training data were randomly re-sampled for each repetition. Then we employed the mean K-L divergence as a measure for this condition (the BN and the data size).

- CBD priors for the CMAP+ method were determined according to equation (15), and $\overline{\theta}$ was approximated by a strictly feasible point.

- Methods using Dirichlet priors shared the same weight of priors, which is $\sum_k(\tau_{ijk} - 1) = r_i$. For the CBD prior of CMAP+, γ_{ij} was set as r_i.

Table 1: Basic information of standard BNs

BNs	Nodes	Arcs	Parameters	Constraints
Andes	223	338	1157	935
Win95pts	76	112	574	222
Hepar2	70	123	1453	398
Hailfinder	56	66	2656	437
Alarm	37	46	509	186
Insurance	27	52	984	306
Boerlage92	23	36	86	73
Sachs	11	17	178	79
Asia	8	8	18	16
Survey	6	6	21	16
Cancer	5	4	10	10
Earthquake	5	4	10	10
Weather	4	4	9	7

In this way, parameter learning methods were compared under different BNs, data sizes, and ratios of constrained parameters (ranging from 0.1645 to 1), making the results

relatively fair. The information for BNs is shown in Table 1, and the learning results are given in Tables 2–4, where the best results are highlighted in bold.

From Tables 2–4, we can draw the following conclusions.

Overall According to the results, the CMAP+ performs the best overall in this group of experiments, while CMAP takes second place. On one hand, the 'MEAN' K-L divergence obtained by CMAP+ is consistently the smallest for the data size ranges 50, 100, 500; CMAP is only outperformed by CMAP+. On the other hand, for 39 learning cases with different BNs and data sizes, CMAP+ achieved the best results in 31 cases, while 9 of the best results are realized by CMAP. However, CML and MAP respectively perform best in one case but never the best for ML.

Using Constraints or Not In contrasting the learning results of ML and CML, as well as those of MAP and CMAP, we find that the learning accuracy is significantly improved by incorporating constraints. For different data sizes, from CML to ML, the 'MEAN' K-L divergences respectively decrease by 35.9%, 30.3%, and 25.4%; from CMAP to MAP, the 'MEAN' K-L divergence respectively decreases by 45.9%, 34.6%, and 38.5%.

Using Dirichlet Priors or Not In comparing the learning results of ML and MAP, as well as those of CML and CMAP, it can be observed that Dirichlet priors are extremely helpful for enhancing CPT learning accuracy. The 'MEAN' K-L divergence achieved by MAP with 100 data is better than that achieved by ML with 500 data. More notably, CMAP using only 50 data outperforms CML using 500 data as its 'MEAN' K-L divergence is 80.2% of that achieved by the latter.

CBD Priors vs. flat Priors In fact, we can roughly guess that the CBD prior is overall better than the flat prior, because it is compatible with expert judgments, whereas the latter is not. That the CMAP+ outperforms CMAP by 8.6% reduction on the 'MEAN' K-L divergence supports this estimation.

Table 2: Learning results on standard BNs with 50 data

BNs	ML	MAP	CML	CMAP	CMAP+
Andes	0.305 ± 0.018	0.146 ± 0.002	0.196 ± 0.015	0.073 ± 0.002	$\mathbf{0.063} \pm 0.002$
Win95pts	0.274 ± 0.023	0.226 ± 0.005	0.189 ± 0.013	0.146 ± 0.001	$\mathbf{0.136} \pm 0.001$
Hepar2	0.251 ± 0.022	0.125 ± 0.002	0.203 ± 0.020	0.109 ± 0.001	$\mathbf{0.095} \pm 0.001$
Hailfinder	0.309 ± 0.012	0.116 ± 0.001	0.272 ± 0.012	0.083 ± 0.001	$\mathbf{0.078} \pm 0.001$
Alarm	0.256 ± 0.017	0.249 ± 0.009	0.089 ± 0.012	0.057 ± 0.002	$\mathbf{0.052} \pm 0.002$
Insurance	0.260 ± 0.020	0.215 ± 0.004	0.160 ± 0.013	0.102 ± 0.002	$\mathbf{0.091} \pm 0.002$
Boerlage92	0.334 ± 0.096	0.067 ± 0.005	0.234 ± 0.050	0.044 ± 0.003	$\mathbf{0.034} \pm 0.004$
Sachs	0.308 ± 0.041	0.162 ± 0.008	0.193 ± 0.039	0.099 ± 0.004	$\mathbf{0.076} \pm 0.003$
Asia	0.224 ± 0.125	0.144 ± 0.020	0.158 ± 0.037	$\mathbf{0.138} \pm 0.017$	0.142 ± 0.018
Survey	0.178 ± 0.089	0.039 ± 0.010	0.154 ± 0.077	$\mathbf{0.032} \pm 0.009$	$\mathbf{0.032} \pm 0.009$
Cancer	0.142 ± 0.105	0.097 ± 0.030	0.066 ± 0.081	0.018 ± 0.003	$\mathbf{0.009} \pm 0.007$
Earthquake	0.367 ± 0.258	0.130 ± 0.032	0.133 ± 0.104	0.019 ± 0.004	$\mathbf{0.011} \pm 0.006$
Weather	0.054 ± 0.022	0.043 ± 0.015	0.050 ± 0.024	$\mathbf{0.025} \pm 0.009$	$\mathbf{0.025} \pm 0.011$
MEAN	0.251	0.135	0.161	0.073	**0.065**

Table 3: Learning results on standard BNs with 100 data

BNs	ML	MAP	CML	CMAP	CMAP+
Andes	0.249 ± 0.013	0.108 ± 0.002	0.164 ± 0.012	0.057 ± 0.001	$\mathbf{0.052} \pm 0.001$
Win95pts	0.278 ± 0.027	0.205 ± 0.005	0.217 ± 0.028	0.141 ± 0.001	$\mathbf{0.132} \pm 0.001$
Hepar2	0.245 ± 0.018	0.115 ± 0.002	0.207 ± 0.015	0.103 ± 0.001	$\mathbf{0.091} \pm 0.001$
Hailfinder	0.297 ± 0.011	0.098 ± 0.002	0.275 ± 0.009	0.075 ± 0.001	$\mathbf{0.071} \pm 0.001$
Alarm	0.224 ± 0.014	0.215 ± 0.007	0.092 ± 0.015	0.053 ± 0.002	$\mathbf{0.047} \pm 0.002$
Insurance	0.208 ± 0.017	0.178 ± 0.004	0.136 ± 0.013	0.089 ± 0.002	$\mathbf{0.079} \pm 0.002$
Boerlage92	0.236 ± 0.072	0.050 ± 0.005	0.164 ± 0.055	0.033 ± 0.004	$\mathbf{0.028} \pm 0.005$
Sachs	0.230 ± 0.041	0.132 ± 0.007	0.143 ± 0.029	0.081 ± 0.005	$\mathbf{0.063} \pm 0.003$
Asia	0.141 ± 0.076	0.102 ± 0.016	0.123 ± 0.071	$\mathbf{0.099} \pm 0.015$	0.103 ± 0.016
Survey	0.158 ± 0.088	0.029 ± 0.009	0.114 ± 0.079	$\mathbf{0.023} \pm 0.006$	0.024 ± 0.006
Cancer	0.239 ± 0.186	0.056 ± 0.020	0.118 ± 0.093	0.013 ± 0.002	$\mathbf{0.009} \pm 0.004$
Earthquake	0.419 ± 0.259	$\mathbf{0.126} \pm 0.031$	0.148 ± 0.090	0.150 ± 0.006	0.140 ± 0.007
Weather	0.023 ± 0.021	0.022 ± 0.008	0.020 ± 0.021	0.016 ± 0.004	$\mathbf{0.014} \pm 0.012$
MEAN	0.227	0.110	0.147	0.072	**0.066**

Table 4: Learning results on standard BNs with 500 data

BNs	ML	MAP	CML	CMAP	CMAP+
Andes	0.118 ± 0.010	0.044 ± 0.002	0.087 ± 0.004	$\mathbf{0.024} \pm 0.001$	0.027 ± 0.001
Win95pts	0.267 ± 0.023	0.159 ± 0.003	0.239 ± 0.023	0.124 ± 0.002	$\mathbf{0.117} \pm 0.002$
Hepar2	0.222 ± 0.012	0.093 ± 0.002	0.198 ± 0.012	0.088 ± 0.001	$\mathbf{0.079} \pm 0.001$
Hailfinder	0.294 ± 0.010	0.058 ± 0.001	0.286 ± 0.009	0.050 ± 0.001	$\mathbf{0.048} \pm 0.001$
Alarm	0.139 ± 0.018	0.136 ± 0.005	0.067 ± 0.012	0.037 ± 0.001	$\mathbf{0.031} \pm 0.001$
Insurance	0.117 ± 0.009	0.098 ± 0.003	0.083 ± 0.008	0.055 ± 0.001	$\mathbf{0.047} \pm 0.001$
Boerlage92	0.062 ± 0.017	0.018 ± 0.003	0.045 ± 0.013	$\mathbf{0.011} \pm 0.002$	$\mathbf{0.011} \pm 0.002$
Sachs	0.125 ± 0.029	0.076 ± 0.005	0.083 ± 0.014	0.046 ± 0.003	$\mathbf{0.037} \pm 0.002$
Asia	0.052 ± 0.025	0.055 ± 0.010	$\mathbf{0.044} \pm 0.019$	0.055 ± 0.009	0.056 ± 0.010
Survey	0.072 ± 0.079	0.013 ± 0.009	0.041 ± 0.059	$\mathbf{0.009} \pm 0.007$	$\mathbf{0.009} \pm 0.007$
Cancer	0.072 ± 0.096	0.022 ± 0.020	0.042 ± 0.052	$\mathbf{0.006} \pm 0.002$	0.010 ± 0.005
Earthquake	0.113 ± 0.068	0.072 ± 0.019	0.059 ± 0.057	$\mathbf{0.010} \pm 0.003$	0.014 ± 0.005
Weather	0.003 ± 0.001	0.004 ± 0.001	0.003 ± 0.001	0.003 ± 0.001	$\mathbf{0.002} \pm 0.001$
MEAN	0.122	0.065	0.091	0.040	**0.038**

5.1.2. Incomplete Data

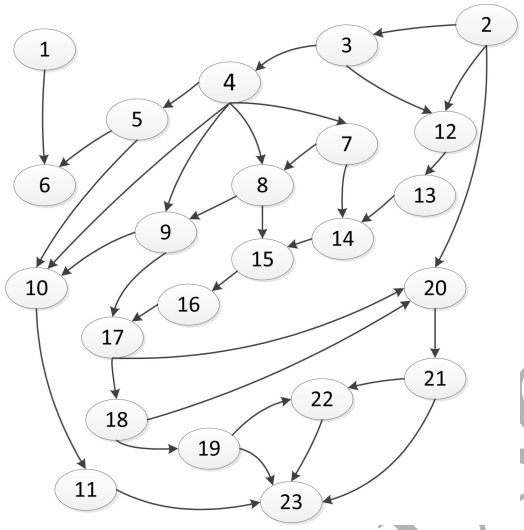


Figure 1: Boerlage92 network with 23 nodes.

To compare the five algorithms under incomplete data, we conducted two experiments on the Boerlage92 network (shown in Figure 1). The experiment settings are summarized as follows:

- The parameter constraints of Boerlage92 network were the same constraints used in the experiments with complete data.
- Incomplete data were obtained by removing samples of hidden nodes from complete data, which were generated based on the true parameters.
- The parameters were learned from incomplete data by combining the five complete-data-driving methods and the EM algorithm, respectively. The hyper-parameters of the five methods were introduced at the beginning of Section 4.
- The K-L divergence was used for measuring the errors between the true parameter and estimated parameter. We used equation (27) to compute the K-L divergence for

a BN. For the K-L divergence of a CPT, we used the follow formula:

$$\overline{KL}(\theta_i, \hat{\theta}_i) = \frac{1}{r_i q_i} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \theta_{ijk} \log(\frac{\theta_{ijk}}{\hat{\theta}_{ijk}})$$

- Two data sizes (500, 1000) and two sets of hidden nodes, (2, 4, 5, 10) and (2, 4, 5, 10, 12, 14, 16, 18), were considered in the experiments.
- Learning was repeated 10 times and training data were randomly re-sampled for each repetition. Then we used the mean of the 10 K-L divergences as a measure for the learning.

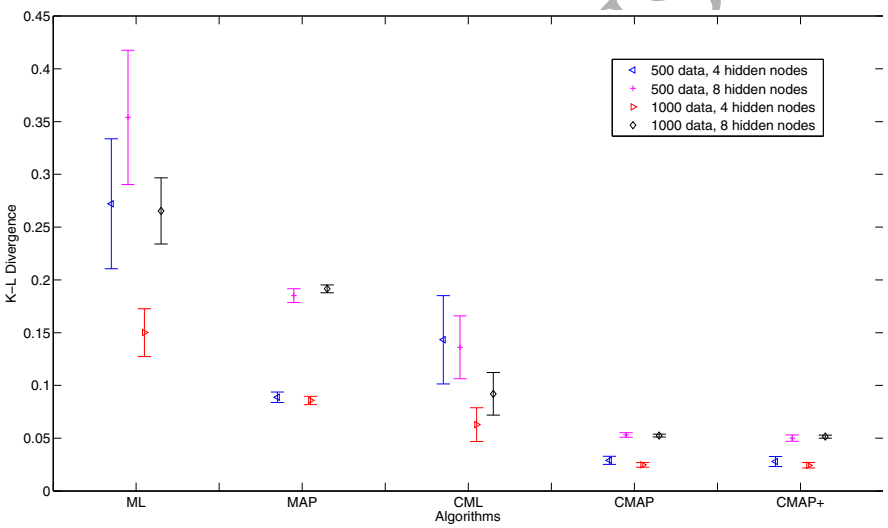


Figure 2: Learning results from incomplete data for the Boerlage92 network. Two data sizes (500, 1000) and two sets for hidden nodes, (2, 4, 5, 10) and (2, 4, 5, 10, 12, 14, 16, 18), were considered.

Experiment 1. In this experiment, we focused on the learning accuracy of the whole Boerlage92 network when the data size and hidden node set varied. The data sizes were set as 500 and 1000, and the hidden nodes were set as (2, 4, 5, 10) and (2, 4, 5, 10, 12, 14, 16, 18). For learning under each data size and hidden node set, the K-L divergences for the whole Boerlage92 network were collected. Figure 2 illustrates the results.

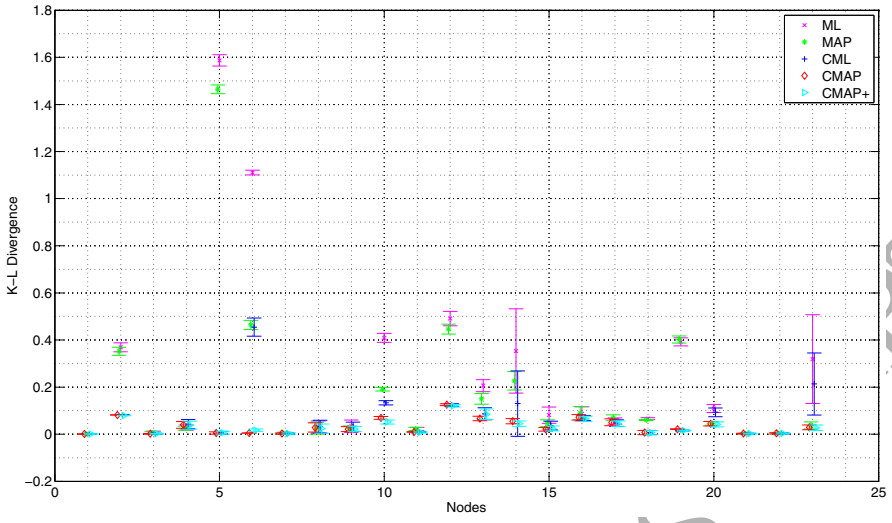
Figure 2 shows that the proposed methods (CMAP and CMAP+) outperformed the other methods under all of the four conditions. In general, with data size increasing and hidden nodes reducing, learning accuracy improved for all algorithms. However, the proposed methods achieved better K-L divergence (≤ 0.053), under the worst condition (500 data and eight hidden nodes), than did the other methods under the best condition. Note that, the K-L divergence of CML is ≥ 0.063 when there were 1000 data and four hidden nodes (the best condition).

Experiment 2. In this experiment, the learning accuracy for each CPT of the Boerlage92 network was concerned. The hidden nodes were fixed at eight (2, 4, 5, 10, 12, 14, 16, 18). For learning under each data size (500 or 1000), the K-L divergences for each CPT were collected. The main results are summarized in Figure 3.

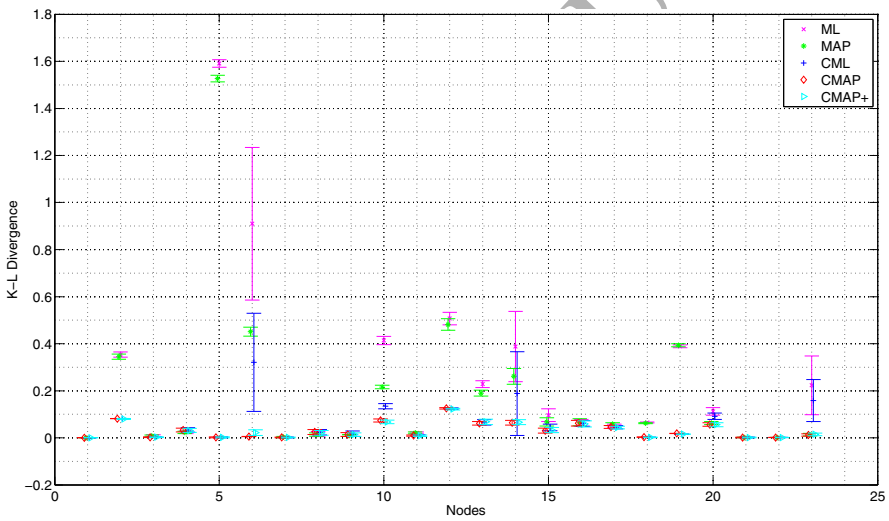
Figure 3 shows that the learning results under two data sizes are close. For nodes 6, 10, and 14, the proposed methods obviously outperformed the other three competing methods. For nodes 2, 5, 12, 13, 18, and 19, the results of CMAP and CMAP+ were better than those of ML and MAP. For nodes 20 and 23, the proposed methods performed better than ML and CML. To sum up, for each node, the accuracies of the proposed methods were the best or close to the best, which explains why they performed best in *Experiment 1.*

5.2. Case Study

To compare the proposed method with the existing algorithms under the condition of scarce and incomplete data, we consider a real facial action unit recognition application from the computer vision domain. According to the Facial Action Unit System (FACS), each AU occurs when the related facial muscles are contracting. FACS is a convenient means to characterize a variety of basic facial expressions by the combination of only a small set of AUs. Thus, although a number of methods have been developed in recent years to directly recognize basic facial expressions, we can also first recognize facial AUs. We then determine the facial expression according to the criteria that facial expressions consist of AUs [36, 37].



(a) 500 data



(b) 1000 data

Figure 3: Learning results for CPTs of the Boerlage92 network. Two data sizes (500, 1000) and eight hidden nodes, (2, 4, 5, 10, 12, 14, 16, 18), were considered.

5.2.1. BN for AU recognition

In practice, it is probably unreliable to respectively recognize each AU by only using current computer vision techniques in scenarios of ambiguity and uncertainty, as well as

under individual differences and dynamic natures of facial actions. Fortunately, there are some inherent relationships among AUs according to the FACS manual [38], which comprise helpful knowledge for overcoming the drawback of respectively recognizing computer vision techniques. Furthermore, the Bayesian network is an appropriate tool to express such knowledge and infer the AUs.

Table 5: Facial Action Units

AUs	Facial Action	AUs	Facial Action	AUs	Facial Action
AU1	Inner brow raiser	AU2	Outer brow raiser	AU4	Brow lowerer
AU5	Upper lid raiser	AU6	Cheek raiser	AU7	Lid tighten
AU9	Nose wrinkle	AU12	Lid corner puller	AU15	Lip corner depressor
AU17	Chin raiser	AU23	Lip tighten	AU24	Lip presser
AU25	Lips part	AU27	Mouth stretch		

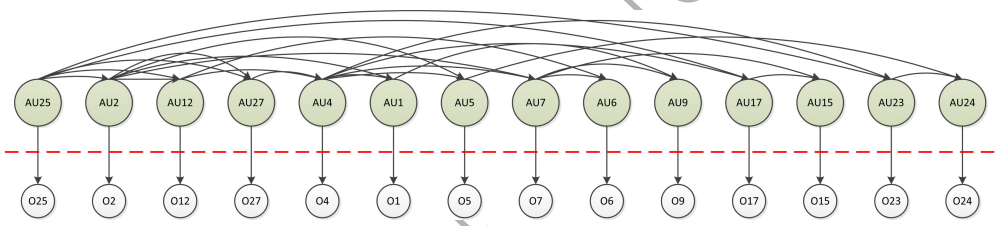


Figure 4: A Bayesian network for AU recognition. The shaded nodes are AUs, which are above the red line. The measurement nodes are beneath the red line.

Instead of recognizing each AU alone, we constructed a BN structure (see Figure 4), including 14 AUs (shown in Table 5), based on the probabilistic relationships among AUs [38]. The structure expresses the mutually exclusive relationships and co-occurrence rules described in the FACS manual. To incorporate the AU recognition results from a computer vision technique, the BN structure introduces a measurement node for each AU. As a result, there are a total of 28 nodes in the structure. Besides the arcs among AUs, each AU has an arc connected to its measurement node. A measurement node does not connect to the other measurement nodes considering that AUs are independently measured. As AUs are not known in advance, they are all set as hidden (unobserved) nodes. On the

contrary, measurement nodes are all observable. Accordingly, the BN can be divided into a measurement layer and an AU layer.

Once CPTs are determined, AU recognition can be performed by running probabilistic inferences on the complete BN.

5.2.2. Learning AU BN with Convex Constraints

To learn the AU BN, an adequate number of unbiased complete training data are required. However, it may be difficult to collect such data in practice. For one, labeling a mass of AUs by domain experts is time-consuming and expensive. Secondly, the reliability of manually labeled AUs is debatable because experts are often confined to ambiguous images or individual differences. In addition, rarely occurring AUs and unfair samples are unavoidable. Thus, biased, scarce, and incomplete training data are often gathered, which can ultimately result in low learning accuracy. As supplementary information, qualitative constraints, implicated in the inherent relationships between AUs, are valuable for improving learning accuracy.

Accordingly, we first introduced qualitative influences on the AU BN: $AU2 \xrightarrow{+} AU1$, $AU4 \xrightarrow{+} AU1$, $AU25 \xrightarrow{+} AU2$, $AU12 \xrightarrow{-} AU4$, $AU27 \xrightarrow{-} AU4$, $AU2 \xrightarrow{+} AU5$, $AU7 \xrightarrow{+} AU6$, $AU12 \xrightarrow{+} AU6$, $AU1 \xrightarrow{-} AU7$, $AU4 \xrightarrow{+} AU7$, $AU1 \xrightarrow{-} AU9$, $AU7 \xrightarrow{+} AU9$, $AU7 \xrightarrow{-} AU15$, $AU17 \xrightarrow{+} AU15$, $AU4 \xrightarrow{+} AU17$, $AU25 \xrightarrow{-} AU17$, $AU25 \xrightarrow{+} AU23$, $AU5 \xrightarrow{-} AU24$, $AU23 \xrightarrow{+} AU24$, $AU2 \xrightarrow{+} AU27$, and $AU25 \xrightarrow{+} AU27$.

In addition to the qualitative influences, we consider four other types of constraints as follows: (a) If AUi has more than one parents and all of them have positive influences, then $p(AUi = 1|\Pi(AUi) = 1) \geq 0.7$, where $\Pi(AUi) = 1$ means all parents are present. (b) Conversely, if AUi has more than one parents and all of them have negative influences, then $p(AUi = 1|\Pi(AUi) = 1) \leq 0.2$. (c) $AU25$ has a relatively small probability of occurring, that is $p(AU27 = 1) \leq 0.5$. (d) We can further give the measurement accuracy, where $p(oi = 1|AUi = 1)$ and $p(oi = 0|AUi = 0)$ can be respectively restricted in a small range.

We employed the EM algorithm integrated with complete-data-driven methods to pa-

rameterize the AU BN from incomplete data and the constraints. The incomplete data included records of measurement nodes but not AU nodes (Figure 4). In addition, the EM procedure started from a set of random CPTs.

5.2.3. Recognition Results

Table 6: Accuracies of SVMs

AU	1	2	4	5	6	7	9
Present	0.596	0.544	0.666	0.539	0.659	0.660	0.681
Absent	1.000	1.000	1.000	1.000	1.000	1.000	1.000
AU	12	15	17	23	24	25	27
Present	0.714	0.716	0.711	0.658	0.691	0.677	0.506
Absent	1.000	1.000♯	1.000♯	1.000	1.000	1.000	1.000

We used the CK+ dataset [36] to test the performance of different parameter learning methods. There are 593 image sequences of images, from neutral to peak frames, across 123 people. The CK+ database provides AUs occurring in each sequence. We labeled the 40% of images close to peak frames with the AUs given by the CK+ database, and the 6% of images close to the neutral frames with empty AU. In this way, we collected more than 5000 labeled images from the CK+ database. A total of 1000 images were used for training and 4000 were used for testing. The training data for each measurement node (observable node) were obtained by a one-vs-all two class linear support vector machine (SVM), which was trained from all neutral and peak frames. The 1000 training images were classified by the SVMs to provide samples for measurement nodes in the AU BN; however, there were no samples for AU nodes (hidden nodes). In such a way, we collected 1000 incomplete training samples. Similarly, we collected 4000 incomplete testing samples (for performing inferences). The accuracies of these SVMs are shown in Table 6 (♯ refers to values close to one). From Table 6, we obtained constraints for the measurement accuracies in the AU BN. That is $p(oi = 0|AUi = 0) \in [0.99, 1]$ and $p(oi = 1|AUi = 1)$ is limited to the 0.025-neighborhood of the true positive accuracy of the related SVM. Then we used the 1000 incomplete samples (AU nodes had no samples) and constraints (including the

constraints mentioned in subsection 4.2.2) to train the AU BN.

After the AU BN had been trained, we inferred the posterior probabilities of each AU given measurement nodes of the AU and its parents (for AU25, as it has no parent, it was inferred based on the measurement nodes of AU25 and its child) by using the junction tree algorithm [39, 40]. If the probability of being present is greater than 0.5, the AU was treated as present; otherwise, the AU was absent. For example, when we infer $p(AU2|o2, o25)$, where $o2$ is the measurement node for AU2, and $o25$ is the measurement node for AU2's parent AU25, if $p(AU2 = 1|o2 = 1, o25 = 0) > 0.5$, then we believe AU2 will be present when $o2 = 1$ and $o25 = 0$. If any of the 4000 test samples (obtained from testing images by the SVMs) satisfies $o2 = 1$ and $o25 = 0$, we believe that AU2 is present in the image. The recognition results were compared with the labels over the 4000 testing images to get the *true positive rate* (successful rate of judging an AU presenting) and *true negative rate* (successful rate of judging an AU absenting). We applied the true positive rate and true negative rate to measure the recognition accuracy, with both being simultaneously higher considered better. The AU recognition results are shown in Figure 5. In addition, Figure 6 illustrates the improvements of true positive and negative rates by combining the AU BN, which are the differences between recognition accuracies of the AU BN and SVMs.

Conclusions drawn from Figures 5&6 are summarized as follows:

- For ten AUs, the algorithms using constraints improved the true positive rate, but the true negative rate decreased for four of the ten AUs.
- The algorithms using constraints achieved higher accuracies than those not using constraints for recognizing all fourteen AUs. For AU25, although ML and MAP show higher true positive rates, they failed to recognize all negative cases.
- The algorithms not using constraints barely improved (or even worsened) the AU recognition accuracy.

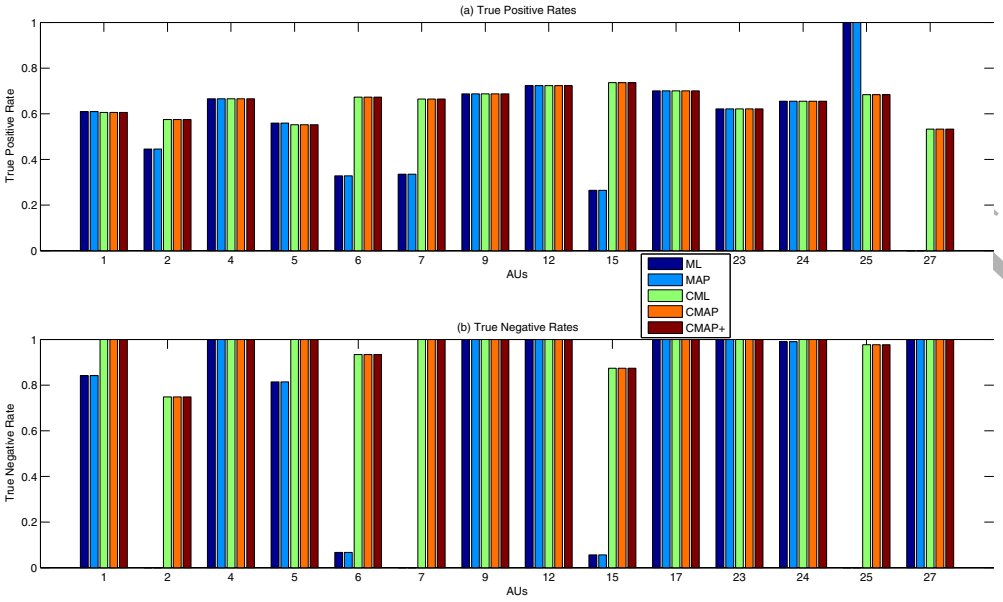


Figure 5: AU recognition results. (a) True positive rates; (b) True negative rates. Higher is better.

- The performances of ML and MAP were close, and the performances of CML, CMAP, and CMAP+ were close.

The first conclusion indicates that the AU BN is helpful for improving the AU recognition accuracy of respective recognition methods, like the one-vs-all linear SVMs if good parameters are learned. The mutually exclusive relationships and co-occurrence rules implicated in the AU BN can provide supplementary information for recognizing an AU, like the measurement nodes of parents. The second and third conclusions declare that incorporating parameter constraints is effective for escaping an undesired local optimum when learning parameters from incomplete data, because the expectation of a likelihood function is multimodal. Although, the AU BN is insensitive to subtle changes, 'significant' changes, such as from 0 to 0.1 or from 0.2 to 0.8, can bring about different results. Using parameter constraints is an effective way to avoid 'significant' changes. Therefore, CML, CMAP, and CMAP+ outperform ML and MAP.

Despite of the same recognition results, the posterior distributions of AU conditioned

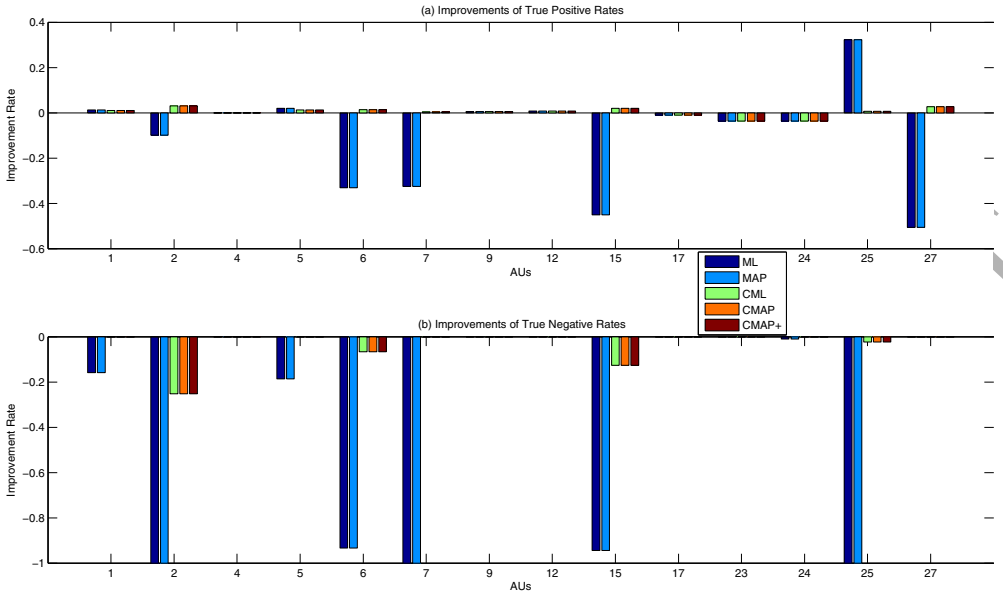


Figure 6: Improvements of true positive and negative rates from SVMs. (a) Improvements of true positive rates; (b) Improvements of true negative rates. Higher is better. A positive value represents the accuracy increasing, while a negative value represents accuracy decreasing.

on some measurement nodes, obtained by different learning algorithms, are not identical. Taking $p(AU2 = 1|o2 = 1, o25 = 0)$ as an example, the inference results of the AU BNs from CML, CMAP, and CMAP+ are respectively 1.00, 0.926, and 0.999. For classification, those algorithms obtain the same conclusion (i.e., AU2 is present). However, a probability of 1.00 is extremely different from 0.926 and 0.999, because it means that the classification result is absolutely correct, which is impossible.

6. Conclusion

In practice, data are often (locally) scarce, which makes it difficult to reveal true CPTs. Moreover, incomplete data can be collected when there are missing records or unobservable variables. The expectation of a likelihood function on incomplete data is multimodal; thus, purely data-driven algorithms may be trapped in undesired local optimums. Therefore, it is often unreliable to parameterize a DAG solely using data.

It has been proved that expert judgments are helpful for improving BN learning accuracy when data are scarce or incomplete. Convex (usually linear) parameter constraints, induced from qualitative expert judgments, can guarantee that estimated CPTs meet domain knowledge. In this paper, we propose a constrained maximum a posteriori approach to learn BN parameters by incorporating convex constraints. In addition to constraints, Dirichlet priors are introduced to alleviate the overfitting problem of the basic convex optimization method. However, widely used BDeu and flat priors are often incompatible with expert judgments, which probably hinders learning accuracy. To further improve the performance of the proposed CMAP algorithm, we introduce a type of constrained Bayesian Dirichlet priors that is compatible with given expert judgments.

A group of experiments were conducted on learning standard BNs from complete and incomplete data. The results show that expert judgments are helpful to enhance CPT learning accuracy when data are limited or incomplete. More importantly, the experimental results demonstrate that the proposed algorithm overall outperforms the methods compared, especially when CBD priors are used. Furthermore, as empirically substantiated by a case study on AU recognition, we conclude that the proposed method can be effectively applied to real-world problems. There are fourteen hidden nodes in the AU BN, and the training data are thus incomplete. From the results, we find that constraints (induced from expert judgments) can improve the recognition accuracy of AU BN; furthermore, we find that most one-vs-all SVMs are improved by using the AU BN.

In this paper we apply a static BN to model dynamic AUs, where time-sensitive information is not considered. However, dynamic BNs can suitably express and infer time-dependent knowledge [41]. Thus, extending the static AU BN to a dynamic BN has the potential to improve recognition accuracy; this will be the focus of our future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61573285).

Appendix A

The interpretation of equation (25). For an incomplete data set $D = \{D_l | l = 1, \cdots, N\}$, let $\overline{D}_l$ be a feasible complete sample of the incomplete sample D_l, and $\overline{\mathbf{D}}_l$ be the set of all feasible complete samples of D_l. Then, the expectation of the likelihood function is computed as

$$\begin{aligned}
&E_{\theta^{(t)}}[\log p(D|\theta)|\theta^{(t)}, D] \\
=&\sum_{l=1}^{N} \sum_{\overline{D}_l \in \overline{\mathbf{D}}_l} p(\overline{D}_l|D_l, \theta^{(t)}) \log p(\overline{D}_l|\theta) \\
=&\sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} (\sum_{l=1}^{N} \sum_{\overline{D}_l \in \overline{\mathbf{D}}_l} p(\overline{D}_l|D_l, \theta^{(t)}) p(X_i = k, \Pi_i = j|\overline{D}_l)) \log \theta_{ijk}
\end{aligned}$$

where $p(\overline{D}_l|D_l, \theta^{(t)})$ is the probability of D_l being $\overline{D}_l$ based on $\theta^{(t)}$, and $p(X_i = k, \Pi_i = j|\overline{D}_l)$ =1 if in $\overline{D}_l$, $X_i = k$ and $\Pi_i = j$, or else $p(X_i = k, \Pi_i = j|\overline{D}_l)$ =0. Thus, we have

$$Q'(\theta|\theta^{(t)}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} (\tau_{ijk} - 1 + \sum_{l=1}^{N} \sum_{\overline{D}_l \in \overline{\mathbf{D}}_l} p(\overline{D}_l|D_l, \theta^{(t)}) p(X_i = k, \Pi_i = j|\overline{D}_l)) \log \theta_{ijk}$$

We can find from the equation that a Dirichlet prior with hyper-parameters $\{\tau_{ijk}\}$ will drive the optimal points of $Q'(\theta|\theta^{(t)})$ to move towards the global optimal of the Dirichlet prior. Thus, a well-defined Dirichlet prior can improve the learning accuracy when data are incomplete.

The astringency of the Algorithm 1. Let $\tau = \{\tau_{ijk}\}$ be the hyper-parameter of a Dirichlet prior $p(\theta)$. Considering that the incomplete sample D_l is included by a complete $\overline{D}_l$ when $\overline{D}_l$ is a possible complete sample of D_l, we have $p(\overline{D}_l|\theta) = p(\overline{D}_l, D_l|\theta)$. In addition,

$\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta)=1$ and $p(D_l|\theta)=\frac{p(\overline{D}_l,D_l|\theta)}{p(\overline{D}_l|D_l,\theta)}$ obviously hold. Thus, we have

$$
\begin{aligned}
&L(\theta|D,\tau)=\log p(D|\theta)p(\theta)\\
=&\log p(\theta)+\log p(D|\theta)\\
=&\log p(\theta)+\sum_l\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log\frac{p(\overline{D}_l,D_l|\theta)}{p(\overline{D}_l|D_l,\theta)}\\
=&\log p(\theta)+\sum_l\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log p(\overline{D}_l|\theta)-\sum_l\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log p(\overline{D}_l|D_l,\theta)\\
=&\log p(\theta)+E_{\theta^{(t)}}[\log p(D|G,\theta)|\theta^{(t)},D]-\sum_l\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log p(\overline{D}_l|D_l,\theta)\\
=&Q'(\theta|\theta^{(t)})-\sum_l\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log p(\overline{D}_l|D_l,\theta)
\end{aligned}
$$

According to the information inequality that the K-L divergence between two distributions is non-negative, we have

$$
\begin{aligned}
&KL(p(\overline{D}_l|D_l,\theta^{(t)}),p(\overline{D}_l|D_l,\theta^{(t+1)}))\\
=&\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log p(\overline{D}_l|D_l,\theta^{(t)})-\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log p(\overline{D}_l|D_l,\theta^{(t+1)})\geq 0
\end{aligned}
$$

That is

$$
\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log p(\overline{D}_l|D_l,\theta^{(t)})\geq\sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)})\log p(\overline{D}_l|D_l,\theta^{(t+1)})
$$

Because $\theta^{(t+1)} = \arg\min_\theta -Q'(\theta|\theta^{(t)})$, $Q'(\theta^{(t)}|\theta^{(t)}) \leq Q'(\theta^{(t+1)}|\theta^{(t)})$ holds. Then we have

$$\begin{aligned} L(\theta^{(t)}|D,\tau) =& Q'(\theta^{(t)}|\theta^{(t)}) - \sum_l \sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t)}) \log p(\overline{D}_l|D_l,\theta^{(t)}) \\ \leq& Q'(\theta^{(t+1)}|\theta^{(t)}) - \sum_l \sum_{\overline{D}_l} p(\overline{D}_l|D_l,\theta^{(t+1)}) \log p(\overline{D}_l|D_l,\theta^{(t+1)}) \\ =& L(\theta^{(t+1)}|D,\tau) \end{aligned}$$

Assuming that $\{\theta^{(t)}|t = 0,1,2,\cdots\}$ is the sequence of estimated parameters obtained by the Algorithm 3, the sequence $\{L(\theta^{(t)}|D,\tau)|t = 0,1,2,\cdots\}$ monotonously increases. As $L(\theta^{(t)}|D,\tau) < 0$, $\{L(\theta^{(t)}|D,\tau)|t = 0,1,2,\cdots\}$ is convergent.

The parameter constraints are not considered in the above discussion, but the result still holds when the mentioned θ, $\theta^{(t)}$, and $\theta^{(t+1)}$ come from a shared constrained domain. To sum up, Algorithm 3 is convergent.

References

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of plausible Inference, Morgan Kaufmann, San Mateo, 1988.

[2] A. C. Constantinou, N. Fenton, W. Marsh, L. Radlinski, From complex questionnaire and interviewing data to intelligent bayesian network models for medical decision support, Artificial Intelligence in Medicine 67 (C) (2016) 75–93.

[3] A. C. Constantinou, M. Freestone, W. Marsh, N. Fenton, J. Coid, Risk assessment and risk management of violent reoffending among prisoners, Expert Systems with Applications 42 (21) (2015) 7511–7529.

[4] B. Yet, Z. Perkins, N. Fenton, N. Tai, W. Marsh, Not just data: A method for improving prediction with knowledge, Journal of Biomedical Informatics 48 (2) (2014) 28C37.

[5] R. Castelo, A. Siebes, Priors on network structures. biasing the search for bayesian networks, International Journal of Approximate Reasoning 24 (1) (2000) 39–57.

[6] N. Fenton, M. Neil, D. A. Lagnado, A general structure for legal arguments about evidence using bayesian networks, Cognitive Science A Multidisciplinary Journal 37 (1) (2013) 61–102.

[7] M. J. Druzdzel, L. C. Van, der Gaag, Elicitation of probabilities for belief networks: Combining qualitative and quantitative information, UU-CS 1995-23 (2013) 141–148.

[8] X. G. Gao, Y. Yang, Z. G. Guo, D. Q. Chen, Bayesian approach to learn bayesian networks using data and constraints, in: International Conference on Pattern Recognition, 2017, pp. 3667–3672.

[9] Y. Zhou, N. Fenton, C. Zhu, An empirical study of bayesian network parameter learning with monotonic influence constraints, Decision Support Systems 87 (C) (2016) 69–79.

[10] C. P. D. Campos, Y. Tong, Q. Ji, Constrained Maximum Likelihood Learning of Bayesian Networks for Facial Action Recognition, Springer Berlin Heidelberg, 2008.

[11] A. Feelders, L. C. V. D. Gaag, Learning bayesian network parameters under order constraints, International Journal of Approximate Reasoning 42 (1-2) (2006) 37–53.

[12] R. Chang, W. Wang, Novel algorithm for bayesian network parameter learning with informative prior constraints, in: Neural Networks (IJCNN), The 2010 International Joint Conference on, 2010, pp. 1–8.

[13] Y. Zhou, N. Fenton, M. Neil, An Extended MPL-C Model for Bayesian Network Parameter Learning with Exterior Constraints, Springer International Publishing, 2014.

[14] W. Liao, Q. Ji, Learning bayesian network parameters under incomplete data with domain knowledge, Pattern Recognition 42 (11) (2009) 3046–3056.

[15] E. E. Altendorf, A. C. Restificar, T. G. Dietterich, Learning from sparse data by exploiting monotonicity constraints, Computer Science (2012) 18–26.

[16] F. Wittig, A. Jameson, Exploiting qualitative knowledge in the learning of conditional probabilities of bayesian networks, in: Conference on Uncertainty in Artificial Intelligence, 2000.

[17] R. S. Niculescu, T. M. Mitchell, R. B. Rao, A theoretical framework for learning bayesian networks with parameter inequality constraints., in: IJCAI 2007, Proceedings of the International Joint Conference on Artificial Intelligence, Hyderabad, India, January, 2007, pp. 155–160.

[18] R. S. Niculescu, T. M. Mitchell, R. B. Rao, Bayesian network learning with parameter constraints., Journal of Machine Learning Research 7 (3) (2006) 1357–1383.

[19] C. P. De Campos, Q. Ji, Improving bayesian network parameter learning using constraints (2008) 1–4.

[20] A. Feelders, L. C. Van, der Gaag, Learning bayesian network parameters with prior knowledge about context-specific qualitative influences, Computer Science (2012) 193–200.

[21] H. D. Brunk, Maximum likelihood estimates of monotone parameters, Annals of Mathematical Statistics 26 (4) (1955) 607–616.

[22] T. Hospedales, Y. Zhou, N. Fenton, M. Neil, Probabilistic graphical models parameter learning with transferred prior and constraints, in: Uncertainty in Artificial Intelligence, 2015.

[23] Y. Zhou, N. Fenton, M. Neil, Bayesian network approach to multinomial parameter learning using data and expert judgments, International Journal of Approximate Reasoning 55 (5) (2014) 1252–1268.

[24] H. Langseth, D. Marquez, M. Neil, Fast Approximate Inference in Hybrid Bayesian Networks Using Dynamic Discretisation, Springer Berlin Heidelberg, 2013.

[25] D. Heckerman, G. Dan, D. M. Chickering, Learning bayesian networks: The combination of knowledge and statistical data, Machine Learning 20 (3) (1995) 197–243.

[26] C. F. J. Wu, On the convergence properties of the em algorithm, Annals of Statistics 11 (1) (1983) 95–103.

[27] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[28] M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in: V. Blondel, S. Boyd, H. Kimura (Eds.), Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110, `http://stanford.edu/~boyd/graph_dcp.html`.

[29] M. Gori, Machine Learning:A Constraint-Based Approach, Morgan Kaufmann, 2017.

[30] H. H. Bauschke, J. M. Borwein, On projection algorithms for solving convex feasibility problems, Siam Review 38 (3) (1996) 367–426.

[31] P. J. Davis, P. Rabinowitz, Methods of numerical integration, Academic Press,, 1975.

[32] I. CVX Research, CVX: Matlab software for disciplined convex programming, version 2.0, `http://cvxr.com/cvx` (Aug. 2012).

[33] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, The MIT Press, 2009.

[34] B. Boerlage, Link strength in Bayesian networks, University of British Columbia, 1992.

[35] S. Kullback, R. A. Leibler, On information and sufficiency, Annals of Mathematical Statistics 22 (22) (1951) 79–86.

[36] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression, in: Computer Vision and Pattern Recognition Workshops, 2010, pp. 94–101.

[37] T. Kanade, Y. Tian, J. F. Cohn, Comprehensive database for facial expression analysis, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2000. Proceedings, 2002, p. 46.

[38] Y. Tong, W. Liao, Q. Ji, Inferring facial action units with causal relations, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 1623–1630.

[39] R. G. Cowell, S. L. Lauritzen, A. P. David, J. Lawless, M. Jordan, Probabilistic Networks and Expert Systems, Springer-Verlag,, 1999.

[40] H. A. Darwiche, Inference in belief networks: A procedural guide, International Journal of Approximate Reasoning 15 (3) (1996) 225–263.

[41] G. Yang, Y. Lin, P. Bhattacharya, A driver fatigue recognition model based on information fusion and dynamic bayesian network, Information Sciences 180 (10) (2010) 1942–1954.

Author Biographies

Yu Yang is a PhD candidate from the Department of System Engineering, Northwestern Polytechnical University, Xian, China. His areas of research include Bayesian networks, data mining, and Image Recognition.

Xiaoguang Gao received the PhD degree from the Northwestern Polytechnical University, Xian, China in 1989. She is currently a professor in the Department of System Engineering, Northwestern Polytechnical University. Her research interests include probabilistic graphical models, deep learning, and reinforcement learning.

Zhigao Guo is currently a PhD candidate at the Department of System Engineering, Northwestern Polytechnical University. His research interests cover Bayesian networks, model optimization, knowledge and data mining. He has been the author of peer-reviewed publications on international journal of Approximate Reasoning, Advanced Methodology for Bayesian Networks, and so on.

Dr. Daqing Chen is a Senior Lecturer and MSc Course Director in the School of Engineering, London South Bank, UK. His main professional expertise and research interests are in the areas of deep learning, business intelligence, and data mining. In recent years, Dr. Chen has led and conducted several business-oriented machine learning and data mining projects across a variety of industry sectors, including Machine-Based Learning for Automatic Construction Cost Indexing (The Royal Institution of Charted Surveyors), Big Data/Fact-driven Decision-Making System (the London Borough of Lambeth), Student Retention Analysis and Prediction (London South Bank University), Donor Segmentation for Not-for-Profit Organization (The Muscular Dystrophy Campaign, UK), and Customer-centric Business Intelligence for Online Retailers (The Rex International, Ltd, UK).