Learning Bayesian Network Parameters via Minimax Algorithm

Xiao-guang Gao[a,*], Zhi-gao Guo[a], Hao Ren[a], Yu Yang[a], Da-qing Chen[b], Chu-chao He[a]

[a] *Department of System Engineering, Northwestern Polytechnical University, China*
[b] *Division of Computer Science and Informatics, London South Bank University, UK*

Abstract

Parameter learning is an important aspect of learning in Bayesian networks. Although the maximum likelihood algorithm is often effective, it suffers from overfitting when there is insufficient data. To address this, prior distributions of model parameters are often imposed. When training a Bayesian network, the parameters of the network are optimized to fit the data. However, imposing prior distributions can reduce the fitness between parameters and data. Therefore, a trade-off is needed between fitting and overfitting. In this study, a new algorithm, named MiniMax Fitness (MMF) is developed to address this problem. The method includes three main steps. First, the maximum a posterior estimation that combines data and prior distribution is derived. Then, the hyper-parameters of the prior distribution are optimized to minimize the fitness between posterior estimation and data. Finally, the order of posterior estimation is checked and adjusted to match the order of the statistical counts from the data. In addition, we introduce an improved constrained maximum entropy method, named Prior Free Constrained Maximum Entropy (PF-CME), to facilitate parameter learning when domain knowledge is provided. Experiments show that the proposed methods outperforms most of existing parameter learning methods.

Keywords: Bayesian network, Overfitting, Minimax algorithm, Maximum entropy

1. Introduction

A Bayesian network (BN) [1] is a joint probability distribution model representing a set of stochastic variables. In particular, a BN consists of a directed acyclic graph that represents the dependent relationship between variables and a numerical section that specifies the conditional probability distribution for each variable. Over the last 30 years since its introduction, BNs have been developed further, and have become a powerful tool with many applications, including fault diagnoses [2], target tracking [3], robot control [4], gene analysis [5], ecosystem modeling [6], signal processing [7], and educational measurement [8]. In general, BN learning can be divided into two parts: structure learning and parameter learning; structure learning involves finding the optimal directed acyclic graph, while parameter learning involves specifying the conditional probability distributions.

In practice, when performing parameter learning, sufficient samples are required, depending on the complexity of the BN. If the data is sufficient, BNs can be easily constructed using traditional methods such as the maximum likelihood (ML) method [9]. However, the ML method tends to overfit when there is insufficient data [10]. Unfortunately, collecting abundant data is a difficult task under certain circumstances, such as, in the cases of earthquake prediction [11], parole assessment [12], and rare disease diagnosis [13]; in such cases, BNs constructed using the ML method will suffer from over-fitting. To avoid over-fitting, prior distributions of model parameters are often imposed. In particular, imposing certain prior distributions will decrease the likelihood of the parameters and therefore reduce the fitness between parameters and data. However, if parameter estimation is tremendously biased toward the prior, the estimation will suffer from underfitting. In this paper, to avoid both over-fitting and under-fitting, we apply the Minimax algorithm to achieve the tradeoff. By applying the Minimax algorithm, we define a novel prior. Unlike the familiar subjective prior, such as flat prior, reference prior, Haldane prior, or Jeffreys prior, the defined prior is objective, which can be written as

$$\theta_{ijk}^{prior} = \begin{cases} 1, & N_{ijk} = \min N_{ijk'}, k^{'} = 1, 2, ..., r_i \\ 0, & N_{ijk} \neq \min N_{ijk'}, k^{'} = 1, 2, ..., r_i, \end{cases}$$

where θ_{ijk}^{prior} denotes the hyper-parameters of prior distribution and N_{ijk} denotes the number of samples, in which the variable X_i adopts the value k and the parent nodes $X_{\pi(i)}$ adopt the configuration state j. The above prior is extreme but makes sense. Maximum likelihood estimation is biased against the least observed event. However, when the available data is limited, maximum likelihood estimation is not trustworthy. Therefore, through the proposed prior, the least observed event becomes less biased and thus the parameter estimation is more acceptable.

Apart from imposing quantitative prior, qualitative domain knowledge also improves parameter estimation. To utilize both data and domain knowledge, we further introduce an improved constrained maximum entropy method. Compared with the traditional constrained maximum entropy method, by our method, domain experts do not need to specify prior strength, which is hard to provide and also has considerable effect on the parameter estimation. The remainder of the paper is organized as follows: In Section 2, the works related to parameter learning and Minimax algorithm application are introduced. A basic discussion of BNs and BN learning is presented in Section 3. In Section 4, the details of the proposed methods are described. In Section 5, a set of experiments are presented to highlight the performance of the proposed methods. Finally, conclusion and directions for future research are presented in Section 6.

2. Related Works

The methods for BN parameter learning using small data sets can be categorized into two types: constraint-based methods and non-constraint-based methods. As evident, non-constraint-based methods are methods that do not consider

[*] Corresponding author
Email address: cxg2012@nwpu.edu.cn (Xiao-guang Gao)

parameter constraints. Among these non-constraint-based methods, Cooper and Herskovits [14] suggested setting hyper-parameters of prior distribution parameters to be 1, which is referred to as uniform prior distribution. In addition, a type of non-informative prior distribution called Jeffrey's prior [15, 16] was proposed by Harold Jefferys, which sets the hyper-parameters of prior distribution parameters to be 0.5. Isozaki et al. [10] proposed a parameter learning method called minimum free energy method. In this method, a free energy function formed using the Kullback-Leibler divergence and an entropy function are defined; a hyper-parameter called data temperature was used to control the proportion between the Kullback-Leibler divergence and entropy function. Subsequently, the required parameters were calculated.

However, it is common to learn parameters with constraints when the data available is insufficient. Wittig and Jameson [17] defined a violation term and applied it as a penalty term for the log-likelihood function, thus obtaining a modified likelihood function; this modified likelihood function was considered the objective function of the optimization model in their study. Finally, the optimization model was optimized using adaptive probabilistic networks. In another study, Altendorf et al. [18] considered monotonicity constraints. They initialized the parameters using ML estimation first. If all the constraints were satisfied, the ML estimation was considered the final parameter; otherwise, a penalty term was defined to penalize the likelihood function. Finally, the penalized likelihood function was optimized using the gradient descent algorithm. Zhou et al. [19] studied Altendorf's method and suggested that the optimization of the penalized likelihood function using gradient descent algorithm caused unacceptably poor parameter estimation results when the data count was zero or extremely small, and the reason for this was that gradient descent started at a random point. To address this problem, they introduced a flat prior distribution to the penalized likelihood function. Further, Feelders et al. [20] also proposed a parameter learning algorithm that first employed the ML estimation and then elicited parameter orders from the parameter constraints. Finally, the isotonic regression algorithm was applied to regulate the initial parameters and the regulated parameters satisfying the parameter orders were elicited from the parameter constraints. Campos et al. proposed the constrained ML method [21] and the constrained maximum entropy method [22]. The constrained ML method constructed a convex optimization model that maximized the likelihood function subject to the parameter constraints. Next, the optimization model was optimized using the convex optimization methods. The constrained maximum entropy method also constructed an optimization model subject to parameter constraints, and an entropy function combining the prior distribution and the data set was defined. Next, an optimization model containing the entropy function was constructed. As in the case of the constrained ML methods, the model was optimized using the convex optimization algorithm. Chang et al. [23] proposed a qualitative MAP method and method involved sampling a certain number of possible parameters from among the parameter constraints using the rejection-acceptance sampling method. Then, the mean values of the sampled parameters were calculated and considered as prior distribution parameters. The final parameters were calculated by combining the data set and prior distribution parameters. It is important to note, however, that the final parameters may not satisfy the parameter constraints in this case. To address this violation, Guo et al. [24] proposed a further constrained qualitative MAP method. After the original qualitative MAP estimation, an optimization model combining the final parameter and the parameter constraints was constructed; thus, the model ensures that the optimized parameters satisfy the parameter constraints.

In fact, Minimax algorithm has already been applied on parameter estimation of various statistical models. For example, Bickel [25] studied the estimation of the mean of a normal distribution with known variance. Given prior knowledge that the mean lies in a known interval, the Minimax estimation is Bayes with respect to a least favorable prior distribution concentrating on a finite number of points. Zou et al. [26] considered the problem of estimating the parameter n of the binomial distribution under the assumption of both infinite and finite parameter spaces. Furthermore, the Minimax property of some estimators is investigated. Besides, Takimoto et al. [27] used the Minimax strategy for on-line density estimation with a Gaussian of unit variance. Interestingly, Silander et al. [28] applied the Minimax algorithm on the parameter estimation of Bayesian networks. In the paper, Minimax regret algorithm was used to maximize and minimize the regret of parameter distribution for data. However, in that paper, the parameter estimation was formulated as a *non*-Bayesian estimation and no pior was assumed and incorporated into the estimation. In this paper, we apply the Minimax algorithm to maximize and minimize the fitness of posterior estimation to data and introduce a new prior. Experiments show that the proposed prior improves the parameter estimation of Bayesian networks.

3. Preliminaries

3.1. BN

A BN is a joint probability distribution model of stochastic variables $X = (X_1, X_2, \cdots, X_n)$. A BN consists of a directed acyclic graph in which each node corresponds to a stochastic variable and the arcs reflect the qualitative dependence between them. In addition, the network includes conditional probability tables (CPTs). Each element of the CPTs can be represented as $p(X_i|X_{\pi(i)})$ for each node X_i given its parent nodes $X_{\pi(i)}$. In practice, θ_{ijk} is used to represent $p(X_i = k|X_{\pi(i)} = j)$ when node X_i adopts the state k and its parent nodes adopt the configuration state j. For node node i, we assume that it has r_i different states, and its parent nodes have q_i different configuration states. The jth row of the CPTs can be represented as $(\theta_{ij1}, \theta_{ij2}, \cdots, \theta_{ijr_i})$, while the kth column of the CPTs can be represented as $(\theta_{i1k}, \theta_{i2k}, \cdots, \theta_{iq_ik})$.

3.2. Parameter Learning

Parameter learning entails estimating CPT values from a data set for a known structure. In this study, we assume that there is no latent variable and no missing value in the data set. In data set D, let N_{ij} denote the number of samples in D in which the parent nodes $X_{\pi(i)}$ adopt the configuration state j. Let N_{ijk} denote the number of samples in D in which the variable X_i adopts the value k and the parent nodes $X_{\pi(i)}$ adopt the configuration state j. Usually, to facilitate the parameter learning of BNs, two assumptions are often employed, which are stated as follows:

Assumption 1 *For data set D that has N instantiations $(D^{(1)}, ..., D^{(N)})$, the instantiations are assumed to be independent and identically distributed. The counts of observations in the data set $(N_{ij1}, ..., N_{ijr_i})$ follows multinomial distributions so that*

$$P(N_{ij1}..., N_{ijr_i}) = N_{ij}! \prod_{k=1}^{r_i} \frac{\theta_{ijk}^{N_{ijk}}}{N_{ijk}!}, \tag{1}$$

where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Assumption 2 *Parameters* $(\theta_{ij1}, ...\theta_{ijr_i})$ *are assumed to be independent of each other and follows Dirichlet distributions so that*

$$P(\theta_{ij}) = \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_k)}{\prod_{k=1}^{r_i} \Gamma(\alpha_k)} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1}, \tag{2}$$

where $(\alpha_{ij1}, ..., \alpha_{ijr_i})$ *are the hyper-parameters of Dirchlet distributions.* $\Gamma(x)$ *is the Gamma function, which is defined as* $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$.

For parameter learning of BNs, the ML estimation is the most common parameter learning algorithm, which is defined as a maximization problem of the data likelihood, which is written as

$$\log P(D|\theta) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\theta_{ijk}. \tag{3}$$

Therefore, the *maximum likelihood estimation* can be given by

$$\theta_{ijk}^{ML} = \frac{N_{ijk}}{N_{ij}}. \tag{4}$$

However, when there is insufficient data, models constructed by ML estimation usually suffer from overfitting. An effective technique to overcome overfitting is to impose prior distributions on model parameters. Then, the parameter estimation amounts to infer from the posterior distribution, which is a combination of data statistics and prior distributions and can be written as follows:

$$P(\theta|D) = \frac{P(D|\theta) \times P(\theta)}{P(D)}. \tag{5}$$

The log form of the posterior distribution can be expressed as

$$\log P(\theta|D) = \log P(D|\theta) + \log P(\theta) - \log P(D), \tag{6}$$

where $P(D|\theta)$ is the data likelihood function, $P(\theta)$ is the prior distribution function, and $P(D)$ is a constant. As the Dirichlet distribution is a natural conjugate of the multinomial distribution [29], it is the most convenient way to convey the prior on parameters. With a Dirichlet prior, the log form of the prior distribution function can represented as

$$\log P(\theta|G) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} \alpha_{ijk} \log \theta_{ijk} + \log \beta, \tag{7}$$

where β is a constant that is used to normalize the parameters of the prior distribution. Then, the log form of posterior distribution can further written as

$$\log P(\theta|D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} (N_{ijk} + \alpha_{ijk}) \log \theta_{ijk} + c, \tag{8}$$

where $c = -\log P(D) + \log \beta$. Finally, the *maximum a posterior estimation* of parameter θ_{ijk} can be computed as

$$\theta_{ijk}^{MAP} = \frac{N_{ijk} + \alpha_{ijk}}{N_{ij} + \alpha_{ij}}. \tag{9}$$

In the *maximum a posterior estimation*, hyper-parameter α_{ijk} is often called the equivalent sample size corresponding to N_{ijk} [23]. The prior distributions can overcome overfitting, because α_{ijk} increases the data size. However, because of the difficulty in specifying the value of α_{ijk}, α_{ijk} is often manually set to different values. For example, $\alpha_{ijk} = 1$ represents the non-informative uniform prior [14] and $\alpha_{ijk} = 0.5$ represents the non-informative Jeffreys prior [15, 16].

3.3. Inequality Relationship

In general, qualitative statements from domain experts can be translated into one of the following parameter constraints [18, 23]:

(1) Range constraint:

$$0 \leq \alpha_{ijk} \leq \theta_{ijk} \leq \beta_{ijk} \leq 1 \tag{10}$$

It defines the upper and lower bounds of a parameter and it is commonly used in practice. In addition, domain experts find it convenient to provide such constraints.

(2) Intra-distribution constraint:

$$\theta_{ijk} \leq \theta_{ijk'}, \forall k \neq k' \tag{11}$$

It describes the comparative relation between two parameters referring to the same parent configuration node state j with different states k and k' of a child node.

(3) Cross-distribution constraint:

$$\theta_{ijk} \leq \theta_{ij'k}, \forall j \neq j' \tag{12}$$

It defines the comparative relation between two parameters referring to the same child node state k with different parent configuration node states j and j'.

(4) Inter-distribution constraint:

$$\theta_{ijk} \leq \theta_{i'j'k'}, \forall i \neq i', j \neq j', k \neq k' \tag{13}$$

It describes the comparative relation between two parameters referring to different nodes.

(5) Approximate-equality constraint:

$$\theta_{ijk} \approx \theta_{i'j'k'}, \forall i \neq i', j \neq j', k \neq k' \tag{14}$$

It defines the close relation between any two parameters. Because the form of the above-mentioned constraints is not convenient for further calculation, it needs to be transformed into the following form:

$$\mid \theta_{ijk} - \theta_{i'j'k'} \mid \leq \varepsilon, \forall i \neq i', j \neq j', k \neq k' \tag{15}$$

where ε is a very small value.

4. The Methods

4.1. MiniMax Fitness Method

In this paper, to avoid both overfitting and underfitting of estimation to data, we optimize the following model that incorporates the prior

$$\max \min \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} log \theta_{ijk}^{posterior}. \tag{16}$$

The above model can be further written as

$$\max_{\alpha} \min_{\theta_{ijk}^{prior}} \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} log \frac{N_{ijk} + \alpha * \theta_{ijk}^{prior}}{N_{ij} + \alpha}. \tag{17}$$

In the above model, α and θ_{ijk}^{prior} are variables tuning the overall fitness of posterior estimation to data. First, to avoid overfitting of posterior estimation to data, the following model is optimized

$$\min_{\theta_{ijk}^{prior}} \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} log \frac{N_{ijk} + \alpha * \theta_{ijk}^{prior}}{N_{ij} + \alpha}. \tag{18}$$

Theorem 1 The minima of model given by Eq. 18 is

$$\theta_{ijk}^{prior} = \begin{cases} 1, & \text{if } N_{ijk} = \min N_{ijk'}, k^{'} = 1, ..., r_i \\ 0, & \text{if } N_{ijk} \neq \min N_{ijk'}, k^{'} = 1, ..., r_i. \end{cases} \tag{19}$$

Please see the proof of Theorem 1 at Appendix A.

Proposition 1 The Mini-Max Fitness estimation is more efficient than Maximum Likelihood estimation when the equivalent sample size satisfies

$$0 < \alpha < \frac{2N_{ij} \sum\limits_{k} N_{ijk}^2 - 2N_{ij}^2 N_{ijm}}{N_{ij}^2 - \sum\limits_{k} N_{ijk}^2}.$$

Please see the proof of Proposition 1 at Appendix D.

Then, since θ_{ijk}^{prior} has been determined, to avoid underfitting of posterior estimation to data, the following model is optimized

$$\max_{\alpha} \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} log \frac{N_{ijk} + \alpha * \theta_{ijk}^{prior*}}{N_{ij} + \alpha} \tag{20}$$

where θ_{ijk}^{prior*} denotes the optimal value of θ_{ijk}^{prior}, which is determined according to Theorem 1. As function

$$f_1(\alpha) = \sum_{k=1}^{r_i} N_{ijk} log \frac{N_{ijk} + \alpha * \theta_{ijk}^{prior*}}{N_{ij} + \alpha} \tag{21}$$

is a strictly decreasing function (see Appendix C), function

$$f_2(\alpha) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} log \frac{N_{ijk} + \alpha * \theta_{ijk}^{prior*}}{N_{ij} + \alpha} \tag{22}$$

is also a strictly decreasing function. Therefore, model given by Eq. 23 is maximized when the prior strength α is zero. However, in that case, the proposed algorithm evolves into *maximum likelihood estimation*, which has been demonstrated to be overfitting when the available data is insufficient. Therefore, prior strength α ought to be larger than zero, in order to avoid overfitting. In fact, when α takes a greater value, it reduces more over-fitting. To determine the value of α, we determine the α value by cross-validation. After determining θ_{ijk}^{prior} and α, the parameters of posterior distribution can be computed as

$$\theta_{ijk}^{posterior} = \frac{N_{ijk} + \alpha * \theta_{ijk}^{prior}}{N_{ij} + \alpha}. \tag{23}$$

However, applying prior in Eq. 21 may cause the order of the hyper-parameters of posterior distribution to differ from the order of the statistical counts from data. For example, if the statistical counts from data is $(N_{ij1}, N_{ij2}, N_{ij3})$, the order of the statistical counts from data is $N_{ij1} > N_{ij2} > N_{ij3}$. Applying the prior distribution in Eq. 17, the hyper-parameters of posterior distribution will be given by $\frac{N_{ij1}}{N_{ij}+\alpha}, \frac{N_{ij2}}{N_{ij}+\alpha}, \frac{N_{ij3}+\alpha}{N_{ij}+\alpha}$. If $N_{ij3} + \alpha > N_{ij2}$, the order of the hyper-parameters of posterior distribution will differ from the order of the statistical counts from data. To address the above problem, the following algorithm has been proposed, which is referred to as the MiniMax Fitness (MMF) algorithm.

Step 1: Count the numbers of observations N_{ijk} and N_{ij} from the data set and determine the prior θ_{ijk}^{prior} according to Theorem 1.

Step 2: Normalize the prior θ_{ij}^{prior}.

Step 3: Compute the parameter estimation $\theta_{ijk}^{posterior}$ according to Eq. 24.

Step 4: Check the numerical order of $\theta_{ijk}^{posterior}$ and the numerical order of N_{ijk}. If the two orders agree, then stop the algorithm. If not, go to Step 5.

Step 5: Reset the prior by the following rule

$$\theta_{ijk}^{prior} = \begin{cases} 1, & \text{if } \theta_{ijk}^{posterior} = \min \theta_{ijk'}^{posterior}, k' = 1, ..., r_i \\ 0, & \text{if } \theta_{ijk}^{posterior} \neq \min \theta_{ijk'}^{posterior}, k' = 1, ..., r_i. \end{cases}$$

Then, go to step 2.

4.2. Prior Free Constrained Maximum Entropy Method

Minimax Fitness algorithm applies to estimation when there is no parameter constraints. In fact, in some cases, domain experts can provide certain knowledge on unknown parameters added as supplement to the insufficient data. The knowledge could be converted into parameter constraints, which are helpful to improve the parameter estimation. To make full use of sample data and domain knowledge, in this part, we present an improved constrained maximum entropy method, called Prior Free Constrained Maximum Entropy (PF-CME) method. For the constrained maximum entropy estimation[22], prior strength α has to be specified by domain experts on the basis of empirical study or cross-validation. In fact, the overall constrained maximum entropy estimation is very sensitive to the prior strength and changing α has a considerable effect on the estimation. To address that problem, we provide a solution that the prior strength α is optimized as a variable of the model formulated as follows

$$\max_{\alpha, \theta_{ijk}^{prior}} \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk} + \alpha \times \theta_{ijk}^{prior}}{N_{ij} + \alpha} \log \frac{N_{ijk} + \alpha \times \theta_{ijk}^{prior}}{N_{ij} + \alpha} \tag{24}$$
$$\textit{subject to } \Omega(\alpha, \theta_{ij}^{prior}).$$

In the above model, along with prior parameter θ_{ijk}^{prior}, prior strength α is optimized to maximized the overall entropy model. It is worth noting that, apart from prior parameter θ_{ijk}^{prior}, the overall posterior estimation $\frac{N_{ijk}+\alpha\times\theta_{ijk}^{prior}}{N_{ij}+\alpha}$ also oughts to satisfy the parameter constraints.

As the model given by Eq. 27 is non-convex, existing techniques, such as Convex Optimization, fail to solve the above problem. In this paper, we use the Sequential Quadratic Programming (SQP) to optimize the model. Then, the procedure of Prior Free Constrained Maximum Entropy method is summarized as follows:

Step 1: Count the numbers of observations N_{ijk} and N_{ij} from the data set and generate parameter constraints from domain knowledge.

Step 2: Set the initial solution of θ_{ijk}^{prior} using Hit-and-Run sampling method[1] to make sure that the initial solution satisfies the parameter constraints. Besides, set the initial solution of prior strength α as a value greater than zero.

Step 3: Optimized the model given by Eq. 27 and determine the optimal value of θ_{ijk}^{prior} and α.

Step 4: Compute the parameter estimation $\theta_{ijk}^{posterior}$ according to Eq. 26.

5. Experiments

We evaluated the learning performance of the proposed algorithm in terms of learning accuracy and learning efficiency. The learning accuracy was evaluated using the Kullback-Leibler (KL) divergence [30], which indicates the divergence between the learned distribution and the true distribution. The learning efficiency was evaluated using the learning time. Learning algorithms implemented in the experiments are listed in Table 1. Among the algorithms, models in CO, CML, CME and FC-QMAP were optimized by Sedumi solver[2]. We perform experiments on four typical benchmark networks. The size of the networks varies from small, medium, large, to considerably large. The information regarding these networks is listed in Table 2.

Table 1: Algorithms implemented in the experiments

Abbreviation	Full name	Constraint-based (Yes/No)
MMF	MiniMax Fitness	No
CO	Convex Optimization	Yes
CME	Constrained Maximum Entropy	Yes
CML	Constrained Maximum Likelihood	Yes
PF-CME	Prior Free Constrained Maximum Entropy	Yes
MAPu	Maximum A Posterior with a Uniform Prior	No
SNML	Sequential Normalized Maximum Likelihood	No
FC-QMAP	Further Constrained Qualitatively Maximum a Posterior	Yes

[1] http://freesourcecode.net/matlabprojects/59958/uniform-distribution-over-a-convex-polytope-in-matlab

[2] http://cvxr.com/cvx/doc/solver.html

Table 2: Details of BNs used in the experiments

	Asia	Alarm	Hailfinder	Andes
Nodes	8	37	56	223
Edges	8	46	66	338
Parameters	18	509	2656	1157
Size Classification	Small	Medium	Large	Considerably Large

5.1. Experiment Setting

The experiments were performed on an Intel Core i7-3770 CPU, 3.40 GHz, and 16 GB RAM. Data were randomly sampled based on true CPTs using a MATLAB program called *samplebnet*[3] and constraints were generated by the following rules: (1) Range constraints are generated as $[\theta_{ijk}^{lower}, \theta_{ijk}^{upper}]$, where $\theta_{ijk}^{lower} = max(0, \theta_{ijk}^{*} - \tau_1)$, $\theta_{ijk}^{upper} = min(1, \theta_{ijk}^{*} + \tau_2)$, where θ_{ijk}^{*} is the true parameter, and τ_1 and τ_2 are two values around 0.2. (2) Inequality constraints are generated as $\theta_{ij_1k_1} \geq \theta_{ij_2k_2}$ if $(\theta_{ij_1k_1} - \theta_{ij_2k_2}) \geq 0.2$. Therefore, when $j_1 = j_2$ and $k_1 \neq k_2$, the generated constraint is the intra-distribution constraint. When $j_1 \neq j_2$ and $k_1 = k_2$, the constraint is the cross-distribution constraint. (3) Approximate-quality constraints are generated as $\theta_{ijk} \approx \theta_{i'j'k'}$ if $\mid \theta_{ijk} - \theta_{i'j'k'} \mid \leq 0.05$.

5.2. Parameter Learning under Different Sample Sizes

First, we examined the learning performance of different methods using different sample sizes. The sample sizes varied from 50, 100, 150, 200, to 250. In the experiments, the maximum constraint number for each node was set to 5. We perform 50 repeated experiments. The average KL-divergence and running time for different networks are summarized in Tables 3 and 4, respectively.

Table 3: Average KL divergence of different algorithms under different sample sizes

	MAPu	SNML	MMF	CO	CML	CME	FC-QMAP	PF-CME
(a) Asia network								
50	1.410±0.009	1.371±0.007	**1.331±0.003**	0.071±0.011	0.053±0.009	0.113±0.011	0.038±0.000	**0.030±0.003**
100	1.319±0.036	1.298±0.034	**1.278±0.035**	0.038±0.002	0.042±0.002	0.070±0.004	0.033±0.002	**0.024±0.005**
150	1.143±0.187	1.098±0.217	**1.060±0.237**	0.031±0.004	0.037±0.004	0.045±0.006	0.020±0.006	**0.018±0.004**
200	1.077±0.241	1.044±0.263	**1.011±0.284**	0.033±0.010	0.025±0.006	0.026±0.006	0.019±0.004	**0.012±0.002**
250	1.065±0.234	1.034±0.257	**1.004±0.278**	0.031±0.008	0.022±0.002	0.020±0.003	0.017±0.006	**0.010±0.003**
(b) Alarm network								
50	0.429±0.023	0.434±0.021	**0.411±0.014**	0.345±0.014	0.342±0.012	0.334±0.013	0.318±0.015	**0.307±0.016**
100	0.371±0.025	0.377±0.022	**0.358±0.021**	0.299±0.013	0.296±0.012	0.290±0.012	0.239±0.014	**0.218±0.015**
150	0.332±0.026	0.339±0.026	**0.313±0.017**	0.271±0.018	0.268±0.018	0.257±0.017	0.208±0.011	**0.191±0.017**
200	0.294±0.014	0.301±0.024	**0.277±0.017**	0.240±0.010	0.237±0.019	0.209±0.015	0.187±0.015	**0.178±0.016**
250	0.281±0.015	0.287±0.016	**0.264±0.018**	0.232±0.012	0.229±0.017	0.161±0.014	0.165±0.014	**0.157±0.014**
(c) Hailfinder network								
50	0.448±0.016	0.497±0.014	**0.424±0.019**	0.311±0.023	0.312±0.013	0.261±0.017	0.256±0.018	**0.235±0.018**
100	0.326±0.010	0.367±0.010	**0.320±0.015**	0.238±0.012	0.239±0.012	0.242±0.013	0.201±0.012	**0.193±0.015**
150	0.265±0.017	0.300±0.007	**0.264±0.014**	0.204±0.011	0.205±0.012	0.196±0.019	0.189±0.016	**0.184±0.019**
200	0.229±0.017	0.259±0.007	**0.233±0.015**	0.184±0.012	0.185±0.013	0.182±0.015	0.162±0.017	**0.146±0.014**
250	0.203±0.017	0.229±0.007	**0.208±0.011**	0.169±0.010	0.170±0.012	0.157±0.013	0.135±0.015	**0.120±0.018**
(d) Andes network								
50	0.118±0.013	0.109±0.019	**0.107±0.017**	0.058±0.005	0.062±0.007	0.071±0.006	0.048±0.008	**0.042±0.005**
100	0.078±0.016	0.071±0.010	**0.070±0.015**	0.040±0.006	0.043±0.005	0.052±0.004	0.046±0.003	**0.033±0.002**
150	0.059±0.010	0.054±0.013	**0.053±0.011**	0.031±0.004	0.034±0.002	0.046±0.006	0.039±0.005	**0.029±0.004**
200	0.049±0.015	0.045±0.009	**0.044±0.008**	0.027±0.003	0.029±0.004	0.033±0.005	0.032±0.006	**0.024±0.005**
250	0.042±0.012	0.039±0.010	**0.038±0.008**	0.023±0.005	0.025±0.003	0.031±0.003	0.028±0.005	**0.020±0.003**

[3] https://github.com/bayesnet/bnt/tree/master/BNT

Table 4: Running time (seconds) of different algorithms under different sample sizes

	MAPu	SNML	MMF	CO	CML	CME	FC-QMAP	PF-CME
(a) Asia network								
50	**0.000±0.000**	**0.000±0.000**	0.008±0.001	0.137±0.008	0.219±0.007	0.240±0.008	0.155±0.007	0.017±0.006
100	**0.001±0.000**	**0.001±0.000**	0.013±0.000	0.140±0.006	0.222±0.009	0.244±0.010	0.211±0.008	0.019±0.004
150	**0.001±0.000**	**0.001±0.000**	0.017±0.000	0.141±0.006	0.223±0.009	0.245±0.010	0.268±0.009	0.019±0.003
200	**0.001±0.000**	**0.001±0.000**	0.021±0.001	0.140±0.005	0.221±0.007	0.243±0.008	0.321±0.007	0.019±0.003
250	**0.002±0.000**	**0.002±0.000**	0.026±0.001	0.139±0.003	0.222±0.005	0.244±0.006	0.374±0.008	0.020±0.003
(b) Alarm network								
50	**0.000±0.000**	**0.000±0.000**	0.005±0.001	0.028±0.005	0.044±0.008	0.048±0.009	0.068±0.008	0.034±0.006
100	**0.000±0.000**	**0.000±0.000**	0.008±0.001	0.028±0.005	0.044±0.008	0.048±0.009	0.111±0.009	0.034±0.006
150	**0.001±0.000**	**0.001±0.000**	0.011±0.001	0.028±0.004	0.043±0.007	0.047±0.007	0.150±0.008	0.032±0.005
200	**0.001±0.000**	**0.001±0.000**	0.014±0.001	0.028±0.004	0.043±0.007	0.047±0.007	0.190±0.007	0.031±0.005
250	**0.001±0.000**	**0.001±0.000**	0.018±0.002	0.029±0.004	0.043±0.006	0.047±0.006	0.232±0.009	0.032±0.006
(c) Hailfinder network								
50	**0.000±0.000**	**0.000±0.000**	0.004±0.000	0.012±0.001	0.017±0.002	0.019±0.002	0.060±0.005	0.052±0.007
100	**0.000±0.000**	**0.000±0.000**	0.008±0.001	0.013±0.001	0.018±0.002	0.020±0.002	0.105±0.008	0.054±0.009
150	**0.001±0.000**	**0.001±0.000**	0.011±0.001	0.013±0.001	0.019±0.002	0.021±0.002	0.147±0.009	0.055±0.010
200	**0.001±0.000**	**0.001±0.000**	0.015±0.001	0.014±0.001	0.020±0.002	0.022±0.002	0.195±0.010	0.060±0.010
250	**0.001±0.000**	**0.001±0.000**	0.018±0.001	0.014±0.001	0.020±0.001	0.022±0.001	0.235±0.010	0.054±0.008
(d) Andes network								
50	**0.000±0.000**	**0.000±0.000**	0.003±0.004	0.027±0.040	0.042±0.007	0.046±0.008	0.050±0.004	0.068±0.009
100	**0.000±0.000**	**0.000±0.000**	0.004±0.006	0.026±0.039	0.040±0.006	0.044±0.007	0.066±0.006	0.065±0.010
150	**0.000±0.001**	**0.000±0.001**	0.006±0.009	0.026±0.040	0.040±0.006	0.044±0.007	0.085±0.008	0.063±0.009
200	**0.000±0.001**	**0.000±0.001**	0.007±0.011	0.026±0.040	0.040±0.006	0.044±0.007	0.103±0.008	0.062±0.006
250	**0.001±0.001**	**0.001±0.001**	0.009±0.013	0.026±0.040	0.040±0.006	0.044±0.007	0.123±0.009	0.063±0.010

Based on the results shown in Table 3, it can be inferred that, in general, as the number of sample increases, all the parameter learning algorithms obtain better estimation. Among all the non-constraint-based methods (MAPu, SNML and MMF), the presented MMF method is demonstrated to be most competitive. Besides, in most cases, the proposed PF-CME method outperforms other constraint-based algorithms (CO, CML, CME, FC-QMAP). From Table 4, we can observe that the proposed MMF method is slightly more time-consuming than other non-constraint-based methods. This is attributed to the process of cross-validation required by the MMF method. However, the proposed PF-CME method is more efficient than other constraint-based algorithms. This is interpreted that, the PF-CME model (Eq. 27) is optimized by SQP approach while models of the rest algorithms are solved by Sedumi solver, which is less efficient than SQP.

5.3. Parameter Learning under Different Constraint Sizes

We performed a set of experiments under different constraint sizes. The sample size was set to 100. In the experiments, the maximum constraint number for each node was set to 1, 2, 3, 4 and 5. We perform 50 repeated experiments. The average KL divergence and running time for different networks are summarized in Tables 5 and 6, respectively.

Table 5: Average KL divergence of different algorithms under different constraint sizes

	MAPu	SNML	MMF	CO	CML	CME	FC-QMAP	PF-CME
(a) Asia network								
1	1.187±0.150	1.127±0.187	**1.104±0.196**	0.883±0.131	0.886±0.128	0.814±0.162	0.760±0.152	**0.643±0.103**
2	1.187±0.150	1.127±0.187	**1.103±0.196**	0.655±0.125	0.643±0.124	0.616±0.114	0.584±0.117	**0.567±0.101**
3	1.187±0.150	1.127±0.187	**1.103±0.196**	0.517±0.117	0.510±0.101	0.390±0.104	0.383±0.111	**0.350±0.009**
4	1.187±0.150	1.127±0.187	**1.103±0.196**	0.328±0.102	0.320±0.128	0.372±0.131	0.347±0.104	**0.286±0.007**
5	1.187±0.150	1.127±0.187	**1.104±0.196**	0.305±0.116	0.298±0.112	0.301±0.109	0.268±0.105	**0.274±0.009**
(b) Alarm network								
1	0.353±0.009	0.358±0.009	**0.331±0.013**	0.307±0.015	0.306±0.015	0.318±0.024	0.348±0.008	**0.314±0.022**
2	0.353±0.009	0.358±0.009	**0.334±0.012**	0.299±0.013	0.299±0.014	0.308±0.024	0.290±0.001	**0.259±0.025**
3	0.353±0.009	0.358±0.009	**0.335±0.010**	0.289±0.023	0.288±0.023	0.301±0.024	0.281±0.015	**0.246±0.072**
4	0.353±0.009	0.358±0.009	**0.335±0.012**	0.292±0.015	0.290±0.015	0.292±0.026	0.276±0.007	**0.231±0.004**
5	0.353±0.009	0.358±0.009	**0.334±0.017**	0.273±0.017	0.270±0.016	0.284±0.026	0.265±0.012	**0.225±0.023**
(c) Hailfinder network								
1	0.322±0.006	0.364±0.006	**0.311±0.000**	0.234±0.002	0.234±0.002	0.247±0.005	0.229±0.010	**0.220±0.006**
2	0.322±0.006	0.364±0.006	**0.314±0.001**	0.232±0.001	0.232±0.001	0.244±0.009	0.226±0.007	**0.217±0.009**
3	0.322±0.006	0.364±0.006	**0.317±0.007**	0.231±0.000	0.230±0.001	0.243±0.006	0.224±0.009	**0.216±0.004**
4	0.322±0.006	0.364±0.006	**0.312±0.004**	0.230±0.002	0.230±0.003	0.241±0.007	0.223±0.006	**0.214±0.007**
5	0.322±0.006	0.364±0.006	**0.319±0.003**	0.228±0.000	0.229±0.000	0.239±0.008	0.221±0.007	**0.212±0.008**
(d) Andes network								
1	0.256±0.001	0.235±0.000	**0.231±0.001**	0.202±0.003	0.199±0.004	0.167±0.005	0.148±0.006	**0.146±0.005**
2	0.256±0.001	0.235±0.000	**0.229±0.003**	0.184±0.009	0.180±0.009	0.153±0.007	0.135±0.007	**0.131±0.008**
3	0.256±0.001	0.235±0.000	**0.228±0.003**	0.179±0.005	0.173±0.004	0.146±0.000	0.128±0.002	**0.125±0.008**
4	0.256±0.001	0.235±0.000	**0.230±0.000**	0.173±0.007	0.165±0.007	0.139±0.000	0.121±0.004	**0.119±0.007**
5	0.256±0.001	0.235±0.000	**0.230±0.002**	0.168±0.003	0.157±0.002	0.132±0.000	0.115±0.008	**0.113±0.004**

Table 6: Running time (seconds) of different algorithms under different constraint sizes

	MAPu	SNML	MMF	CO	CML	CME	FC-QMAP	PF-CME
(a) Asia network								
1	**0.001±0.000**	**0.001±0.000**	0.012±0.001	0.136±0.009	0.156±0.007	0.147±0.006	0.172±0.005	0.014±0.007
2	**0.001±0.000**	**0.001±0.000**	0.012±0.000	0.137±0.006	0.177±0.007	0.153±0.009	0.192±0.008	0.016±0.003
3	**0.001±0.000**	**0.001±0.000**	0.012±0.000	0.138±0.005	0.195±0.008	0.166±0.007	0.205±0.006	0.016±0.003
4	**0.001±0.000**	**0.001±0.000**	0.012±0.000	0.138±0.005	0.208±0.009	0.179±0.005	0.206±0.006	0.016±0.003
5	**0.001±0.000**	**0.001±0.000**	0.012±0.000	0.140±0.005	0.221±0.007	0.196±0.007	0.213±0.004	0.017±0.003
(b) Alarm network								
1	**0.000±0.000**	**0.000±0.000**	0.007±0.000	0.025±0.007	0.029±0.007	0.034±0.007	0.092±0.002	0.017±0.004
2	**0.000±0.000**	**0.000±0.000**	0.007±0.000	0.026±0.009	0.032±0.008	0.039±0.009	0.096±0.002	0.018±0.004
3	**0.000±0.000**	**0.000±0.000**	0.008±0.000	0.026±0.004	0.035±0.005	0.041±0.009	0.100±0.003	0.020±0.005
4	**0.000±0.000**	**0.000±0.000**	0.007±0.000	0.026±0.007	0.038±0.004	0.045±0.006	0.103±0.003	0.020±0.005
5	**0.000±0.000**	**0.000±0.000**	0.007±0.000	0.026±0.005	0.040±0.009	0.049±0.008	0.105±0.002	0.021±0.004
(c) Hailfinder network								
1	**0.000±0.000**	**0.000±0.000**	0.004±0.000	0.006±0.001	0.006±0.001	0.006±0.000	0.047±0.013	0.010±0.001
2	**0.000±0.000**	**0.000±0.000**	0.004±0.000	0.006±0.001	0.007±0.000	0.007±0.001	0.048±0.014	0.010±0.001
3	**0.000±0.000**	**0.000±0.000**	0.004±0.000	0.006±0.000	0.007±0.000	0.007±0.000	0.049±0.015	0.010±0.002
4	**0.000±0.000**	**0.000±0.000**	0.004±0.000	0.006±0.000	0.007±0.001	0.007±0.001	0.049±0.016	0.011±0.001
5	**0.000±0.000**	**0.000±0.000**	0.004±0.001	0.007±0.001	0.008±0.000	0.008±0.000	0.048±0.016	0.010±0.002
(d) Andes network								
1	**0.001±0.000**	**0.001±0.000**	0.011±0.002	0.072±0.007	0.062±0.007	0.065±0.008	0.154±0.011	0.011±0.001
2	**0.001±0.000**	**0.001±0.000**	0.011±0.001	0.081±0.004	0.064±0.006	0.074±0.005	0.166±0.010	0.011±0.001
3	**0.001±0.000**	**0.001±0.000**	0.011±0.001	0.088±0.008	0.066±0.004	0.077±0.007	0.171±0.013	0.012±0.001
4	**0.001±0.000**	**0.001±0.000**	0.011±0.002	0.092±0.009	0.073±0.008	0.080±0.004	0.177±0.009	0.012±0.001
5	**0.001±0.000**	**0.001±0.000**	0.012±0.001	0.097±0.005	0.074±0.004	0.085±0.010	0.183±0.012	0.013±0.001

Based on the results in Table 5, we can infer that, with the increase of parameter constraints, the estimation of constraint-based methods were significantly improved while non-constraint-based methods were less affected. In addition, MMF and PF-CME methods outperformed the other non-constraint-based and constraint-based methods, respectively. It is worth noting that, for networks with higher in-degree or more parameters per node, such as Hailfinder network ($\frac{2656}{56} = 47.4$ parameters per node), slight increase of parameter constraints has minor impact on the parameter estimation. From Table 6, we can draw the conclusion that, with increasing constraints, the time consumption of most non-constraint-based methods remained unchanged while that of constraint-based methods increased by different degrees. This can be interpreted as, with more parameter constraints, models of constraint-based methods become more complex and thus it is more time-consuming to optimize the models.

6. Conclusions

In this study, we described the parameter learning problem in Bayesian networks and proposed a method to address the overfitting problem when the available data is insufficient. Though the imposition of prior distributions on model parameters is helpful in overcoming overfitting, it also may lead to the parameters being less fitted to the data. This subsequently affects parameter learning, wherein parameters are expected to fit the data. The Minimax algorithm is an effective tool to balance fitting and overfitting. Based on the Minimax algorithm, we presented a Minimax Fitness (MMF) algorithm. In addition, to utilize the domain knowledge, we also introduced an improved constrained maximum entropy method, called Prior Free Constrained Maximum Entropy (PF-CME) method. Experiments on several benchmark networks show that the proposed MMF method outperforms mainstream non-constraint-based parameter learning algorithms and PF-CME method outperforms most of the constraint-based learning methods.

For BN parameter learning, methods that does not require specification of prior strength or equivalent sample size deserve more concern and investigation. In the future, we will further improve the proposed PF-CME method by defining more constraints on prior strength.

Acknowledgements

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

References

[1] P. Judea, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, California, 1988.

[2] W. Ibrahim, V. Beiu, Using bayesian networks to accurately calculate the reliability of complementary metal oxide semiconductor gates, IEEE Transactions on Reliability 60 (3) (2011) 538–549.

[3] S. Mascaro, A. E. Nicholso, K. B. Korb, Anomaly detection in vessel tracks using bayesian networks, International Journal of Approximate Reasoning 55 (1) (2014) 84–98.

[4] G. Infantes, M. Ghallab, F. Ingrand, Learning the behavior model of a robot, Autonomous Robots 30 (2) (2011) 157–177.

[5] Y. Tamada, S. Imoto, H. Araki, M. Nagasaki, C. Print, D. S. Charnock-Jones, S. Miyano, Estimating genome-wide gene networks using nonparametric bayesian network models on massively parallel computers, IEEE/ACM Transactions on Computational Biology and Bioinformatics 8 (3) (2011) 683–697.

[6] D. Landuyt, S. Broekx, R. D'hondt, G. Engelen, J. Aertsens, P. L. Goethals, A review of bayesian belief networks in ecosystem service modelling, Environmental Modelling & Software 46 (2013) 1–11.

[7] N. Wachowski, M. R. Azimi-Sadjadi, Detection and classification of nonstationary transient signals using sparse approximations and bayesian networks, IEEE/ACM Transactions on Audio, Speech and Language Processing 22 (12) (2014) 1750–1764.

[8] R. G. Almond, R. J. Mislevy, L. S. Steinberg, D. Yan, D. M. Williamson, Bayesian Networks in Educational Assessment, Springer, 2015.

[9] R. A. Redner, H. F. Walker, Mixture densities, maximum likelihood and the em algorithm, SIAM Review 26 (2) (1984) 195–239.

[10] T. Isozaki, N. Kato, M. Ueno, "data temperature" in minimum free energies for parameter learning of bayesian networks, International Journal on Artificial Intelligence Tools 18 (05) (2009) 653–671.

[11] J. Hu, X. Tang, J. Qiu, A bayesian network approach for predicting seismic liquefaction based on interpretive structural modeling, Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards 9 (3) (2015) 200–217.

[12] 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.

[13] F. L. Seixas, B. Zadrozny, J. Laks, A. Conci, D. C. M. Saade, A bayesian network decision model for supporting the diagnosis of dementia, alzheimer s disease and mild cognitive impairment, Computers in Biology and Medicine 51 (2014) 140–158.

[14] G. F. Cooper, E. Herskovits, A bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (4) (1992) 309–347.

[15] B. S. Clarke, A. R. Barron, Jeffreys' prior is asymptotically least favorable under entropy risk, Journal of Statistical planning and Inference 41 (1) (1994) 37–60.

[16] J. Suzuki, Learning bayesian belief networks based on the minimum description length principle: Basic properties, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 82 (10) (1999) 2237–2245.

[17] F. Wittig, A. Jameson, Exploiting qualitative knowledge in the learning of conditional probabilities of bayesian networks, in: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 2000, pp. 644–652.

[18] E. E. Altendorf, A. C. Restificar, T. G. Dietterich, Learning from sparse data by exploiting monotonicity constraints, in: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2005, pp. 18–26.

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

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

[21] C. P. De Campos, Y. Tong, Q. Ji, Constrained maximum likelihood learning of bayesian networks for facial action recognition, in: Proceedings of the 10th European Conference on Computer Vision, Springer, 2008, pp. 168–181.

[22] C. P. De Campos, Q. Ji, Improving bayesian network parameter learning using constraints, in: Proceedings of the 19th International Conference on Pattern Recognition, IEEE, 2008, pp. 1–4.

[23] R. Chang, W. Wang, Novel algorithm for bayesian network parameter learning with informative prior constraints, in: Proceedings of the International Joint Conference on Neural Networks, IEEE, 2010, pp. 1–8.

[24] Z. Guo, X. Gao, H. Ren, Y. Yang, R. Di, D. Chen, Learning bayesian network parameters from small data sets: A further constrained qualitatively maximum a posteriori method, International Journal of Approximate Reasoning 91 (2017) 22–35.

[25] J. Bickel, P, Minimax estimation of the mean of a normal distribution when the parameter space is restricted, The Annals of Statistics 9 (6) (1981) 1301–1309.

[26] Z. Guohua, W. Alan, Admissible and minimax estimation of the parameter n in the binomial distribution, Journal of Statistical Planning and Inference 113 (2) (2003) 451–466.

[27] T. Eiji, W. Manfred, The minimax strategy for gaussian density estimation, in: Proceedings of the 13th Annual Conference on Computer Learning Theory, Morgan Kaufmann, 2000, pp. 100–106.

[28] S. Tomi, R. Teemu, M. Petri, Locally minimax optimal predictive modeling with bayesian networks, in: Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics, Microtome Publishing, 2009, pp. 504–511.

[29] D. J. Spiegelhalter, S. L. Lauritzen, Sequential updating of conditional probabilities on directed graphical structures, Networks 20 (5) (1990) 579–605.

[30] S. Kullback, R. A. Leibler, On information and sufficiency, The Annals of Mathematical Statistics 22 (1) (1951) 79–86.

Appendix A

Theorem 1. The model

$$\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log \frac{N_{ijk}+\alpha * \theta_{ijk}^{prior}}{N_{ij}+\alpha}$$

is minimized at

$$\theta_{ijk}^{prior} = \begin{cases} 1, & \text{if } N_{ijk} = \min N_{ijk'}, k^{'} = 1, ..., r_i \\ 0, & \text{if } N_{ijk} \neq \min N_{ijk'}, k^{'} = 1, ..., r_i. \end{cases}$$

Proof. (1) First, we assume that $N_{ij1} = \min N_{ijk'}, k^{'} = 1, ..., r_i$. According to global parameter and local parameter independence, the minimization of model is proportional to the minimization of following model

$$\sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}+\alpha * \theta_{ijk}^{prior}}{N_{ij}+\alpha}.$$

Suppose that we have two priors

$$\theta_{ij}^{prior1} = (\theta_{ij1}^{prior1}, \theta_{ij2}^{prior1}, \theta_{ij3}^{prior1}, \cdots, \theta_{ijr_i}^{prior1})$$

and

$$\theta_{ij}^{prior2} = (\theta_{ij1}^{prior2}, 0, \theta_{ij3}^{prior2}, \cdots, \theta_{ijr_i}^{prior2}),$$

where $\theta_{ijk}^{prior1} = \theta_{ijk}^{prior2}, k = 3, 4, \cdots, r_i$. As the above two priors satisfy $\sum_{k=1}^{r_i} \theta_{ijk}^{prior1} = \sum_{k=1}^{r_i} \theta_{ijk}^{prior2} = 1$, we have

$$\theta_{ij1}^{prior2} = \theta_{ij1}^{prior1} + \theta_{ij2}^{prior1},$$

which means

$$\theta_{ij1}^{prior2} > \theta_{ij1}^{prior1}.$$

For simplicity, we define $f(\theta_{ij}^{prior}) = \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}+\alpha*\theta_{ijk}^{prior}}{N_{ij}+\alpha}$. Then,

$$\begin{aligned}
f(\theta_{ij}^{prior2}) - f(\theta_{ij}^{prior1}) &= \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}+\alpha * \theta_{ijk}^{prior2}}{N_{ij}+\alpha} - \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}+\alpha * \theta_{ijk}^{prior1}}{N_{ij}+\alpha} \\
&= N_{ij1} \log \frac{N_{ij1}+\alpha * \theta_{ij1}^{prior2}}{N_{ij}+\alpha} - N_{ij1} \log \frac{N_{ij1}+\alpha * \theta_{ij1}^{prior1}}{N_{ij}+\alpha} + N_{ij2} \log \frac{N_{ij2}}{N_{ij}+\alpha} - N_{ij2} \log \frac{N_{ij2}+\alpha \times \theta_{ij2}^{prior1}}{N_{ij}+\alpha} \\
&= N_{ij1} \log \frac{N_{ij1}+\alpha * \theta_{ij1}^{prior2}}{N_{ij1}+\alpha * \theta_{ij1}^{prior1}} - N_{ij2} \log \frac{N_{ij2}+\alpha * \theta_{ij2}^{prior1}}{N_{ij2}} \\
&= N_{ij1} \log (1 + \frac{\alpha * (\theta_{ij1}^{prior2} - \theta_{ij1}^{prior1})}{N_{ij1}+\alpha * \theta_{ij1}^{prior1}}) - N_{ij2} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij2}}) \\
&= N_{ij1} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij1}+\alpha * \theta_{ij1}^{prior1}}) - N_{ij2} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij2}}) \\
&< N_{ij1} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij1}}) - N_{ij2} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij2}}) \\
&= \alpha * \theta_{ij2}^{prior1} * [\frac{N_{ij1}}{\alpha * \theta_{ij2}^{prior1}} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij1}}) - \frac{N_{ij2}}{\alpha * \theta_{ij2}^{prior1}} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij2}})]
\end{aligned}$$

As function $g(x) = x * \log(1 + \frac{1}{x})$ is an increasing function (see Appendix B) and $\frac{N_{ij1}}{\alpha*\theta_{ij2}^{prior1}} < \frac{N_{ij2}}{\alpha*\theta_{ij2}^{prior1}}$, we have

$$\frac{N_{ij1}}{\alpha * \theta_{ij2}^{prior1}} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij1}}) < \frac{N_{ij2}}{\alpha * \theta_{ij2}^{prior1}} \log (1 + \frac{\alpha * \theta_{ij2}^{prior1}}{N_{ij2}}),$$

which means

$$f(\theta_{ij}^{prior2}) < f(\theta_{ij}^{prior1}).$$

Likewise, for another pair of priors

$$\theta_{ij}^{prior\_2} = (\theta_{ij1}^{prior2}, 0, \theta_{ij3}^{prior2}, \theta_{ij4}^{prior2} \cdots, \theta_{ijr_i}^{prior2}),$$

and

$$\theta_{ij}^{prior\_3} = (\theta_{ij1}^{prior3}, 0, 0, \theta_{ij4}^{prior3}, \cdots, \theta_{ijr_i}^{prior3}),$$

where $\theta_{ijk}^{prior1} = \theta_{ijk}^{prior2}, k = 4, 5, \cdots, r_i$, we can prove that

$$f(\theta_{ij}^{prior3}) < f(\theta_{ij}^{prior2}).$$

Finally, we can prove that, for priors

$$\theta_{ij}^{prior(r_i-1)} = (\theta_{ij1}^{prior(r_i-1)}, 0, 0 \cdots, 0, \theta_{ijr_i}^{prior(r_i-1)}),$$

and

$$\theta_{ij}^{priorr_i} = (1, 0, 0, \cdots, 0),$$

we have

$$f(\theta_{ij}^{prior(r_i)}) < f(\theta_{ij}^{prior(r_i-1)}).$$

Now, we prove that, if $N_{ij1} = \min N_{ijk'}, k^{'} = 1, ..., r_i$, prior $(1, 0, 0, \cdots, 0)$ minimizes the model.

(2) Algebraically, we can prove that, prior

$$\theta_{ijk}^{prior} = \begin{cases} 1, & \text{if } N_{ijk} = \min N_{ijk'}, k^{'} = 1, ..., r_i \\ 0, & \text{if } N_{ijk} \neq \min N_{ijk'}, k^{'} = 1, ..., r_i \end{cases}$$

minimizes the model. □

Appendix B

Proposition 1. Function

$$g(x) = x \times \log(1 + \frac{1}{x})$$

is an increasing function.

Proof. Function $g(x)$ has definition domains $(-\infty, -1)$ and $(0, \infty)$. And, the first and second derivatives of function $g(x)$ are

$$g(x)^{'} = \log(1 + \frac{1}{x}) - \frac{1}{x+1}$$
$$g(x)^{''} = -\frac{1}{x(x+1)^2}.$$

(1) In the definition domain $(-\infty, -1)$, as $g(x)^{''} > 0$, $g(x)^{'}$ is an increasing function. And because

$$\lim_{x \to -\infty} g(x)' = 0,$$

therefore, $g(x)^{'} > 0$ and $g(x)$ is an increasing function in the domain $(-\infty, -1)$.

(2) In the definition domain $(0, \infty)$, as $g(x)^{''} < 0$, $g(x)^{'}$ is a decreasing function. And because

$$\lim_{x \to \infty} g(x)' = 0,$$

therefore, $g(x)^{'} > 0$ and $g(x)$ is an increasing function in the domain $(0, \infty)$.

Appendix C

Proposition 2. Function

$$f(\alpha) = \sum_{k=1}^{r_i} N_{ijk} log \frac{N_{ijk} + \alpha * \theta_{ijk}^{prior}}{N_{ij} + \alpha},$$

where

$$\theta_{ijk}^{prior} = \begin{cases} 1, & \text{if } N_{ijk} = \min N_{ijk'}, k^{'} = 1, ..., r_i \\ 0, & \text{if } N_{ijk} \neq \min N_{ijk'}, k^{'} = 1, ..., r_i \end{cases}$$

is a decreasing function.

Proof. Suppose that $N_{ijK} = \min N_{ijk'}, k^{'} = 1, ..., r_i$, then, we have

$$\theta_{ijk}^{prior} = \begin{cases} 1, & \text{if } k = K \\ 0, & \text{if } k \neq K. \end{cases}$$

Thus, function $f(\alpha)$ can be further written as

$$\begin{aligned} f(\alpha) &= \sum_{k=1}^{r_i} N_{ijk} \log(N_{ijk} + \alpha * \theta_{ijk}^{prior}) - \sum_{k=1}^{r_i} N_{ijk} \log(N_{ij} + \alpha) \\ &= \sum_{k=1, k \neq K}^{r_i} N_{ijk} \log N_{ijk} + N_{ijK} \log(N_{ijK} + \alpha) - N_{ij} \log(N_{ij} + \alpha). \end{aligned}$$

Then, the first derivative of function $f(\alpha)$ is

$$\begin{aligned} f'(\alpha) &= \frac{N_{ijK}}{N_{ijK} + \alpha} - \frac{N_{ij}}{N_{ij} + \alpha} \\ &= \frac{(N_{ijK} - N_{ij})\alpha}{(N_{ijK} + \alpha)(N_{ij} + \alpha)}, \end{aligned}$$

which means, $f'(\alpha) < 0$. Therefore, $f(\alpha)$ is a decreasing function.

Appendix D

Proposition 3 The Mini-Max Fitness estimation is more efficient than Maximum Likelihood estimation when the equivalent sample size satisfies

$$0 < \alpha < \frac{2N_{ij}\sum\limits_k N_{ijk}^2 - 2N_{ij}^2 N_{ijm}}{N_{ij}^2 - \sum\limits_k N_{ijk}^2}.$$

Proof. The variance of Maximum Likelihood estimation is computed as

$$\begin{aligned} D(\frac{N_{ijk}}{N_{ij}}) &= \frac{1}{n}(E(\frac{N_{ijk}^2}{N_{ij}^2}) - (E(\frac{N_{ijk}}{N_{ij}}))^2) \\ &= \frac{1}{n}(E(\frac{N_{ijk}^2}{N_{ij}^2}) - \frac{1}{r_i^2}) \end{aligned}$$

.

The variance of Max-Min Fitness estimation is computed as

$$\begin{aligned} D(\frac{N_{ijk} + \alpha\theta_{ijk}^*}{N_{ij} + \alpha}) &= \frac{1}{n}(E(\frac{(N_{ijk} + \alpha\theta_{ijk}^*)^2}{(N_{ij} + \alpha)^2}) - (E(\frac{N_{ijk} + \alpha\theta_{ijk}^*}{N_{ij} + \alpha}))^2) \\ &= \frac{1}{n}(E(\frac{(N_{ijk} + \alpha\theta_{ijk}^*)^2}{(N_{ij} + \alpha)^2}) - \frac{1}{r_i^2}) \end{aligned}$$

.

We assume N_{ijm} is the minimum observation among $N_{ijk}, (k = 1, 2, ...r_i)$, then, the difference between the variance of Max-Min Fitness estimation and the variance of Maximum Likelihood estimation is computed as

$$\begin{aligned} & D(\frac{N_{ijk} + \alpha\theta_{ijk}^*}{N_{ij} + \alpha}) - D(\frac{N_{ijk}}{N_{ij}}) \\ &= \frac{1}{N_{ij}} E(\frac{(N_{ijk} + \alpha\theta_{ijk}^*)^2}{(N_{ij} + \alpha)^2} - \frac{N_{ijk}^2}{N_{ij}^2}) \\ &= \frac{1}{N_{ij}^2} \sum_k (\frac{(N_{ijk} + \alpha\theta_{ijk}^*)^2}{(N_{ij} + \alpha)^2} - \frac{N_{ijk}^2}{N_{ij}^2}) \\ &= \frac{1}{N_{ij}^2} (\frac{(N_{ijm} + \alpha)^2}{(N_{ij} + \alpha)^2} + \sum_{k, k\neq m} \frac{N_{ijk}^2}{(N_{ij} + \alpha)^2} - \sum_k \frac{N_{ijk}^2}{N_{ij}^4}) \\ &= \frac{(N_{ijm} + \alpha)^2 N_{ij}^2 + N_{ij}^2 \sum\limits_{k, k\neq m} N_{ijk}^2 - (N_{ij} + \alpha)^2 \sum\limits_k N_{ijk}^2}{(N_{ij} + \alpha)^2 N_{ij}^4} \\ &= \frac{N_{ij}^2 N_{ijm}^2 + 2N_{ij}^2 N_{ijm}\alpha + N_{ij}^2\alpha^2 + N_{ij}^2 \sum\limits_{k, k\neq m} N_{ijk}^2 - (N_{ij}^2 \sum\limits_k N_{ijk}^2 + 2N_{ij}\alpha \sum\limits_k N_{ijk}^2 + \alpha^2 \sum\limits_k N_{ijk}^2)}{(N_{ij} + \alpha)^2 N_{ij}^4} \\ &= \frac{(N_{ij}^2 - \sum_k N_{ijk}^2)\alpha^2 + (2N_{ij}^2 N_{ijm} - 2N_{ij} \sum_k N_{ijk}^2)\alpha}{(N_{ij} + \alpha)^2 N_{ij}^4} \\ &= \frac{\alpha}{(N_{ij} + \alpha)^2 N_{ij}^4} ((N_{ij}^2 - \sum_k N_{ijk}^2)\alpha + (2N_{ij}^2 N_{ijm} - 2N_{ij} \sum_k N_{ijk}^2)) \end{aligned}$$

Since $\frac{\alpha}{(N_{ij}+\alpha)^2 N_{ij}^4} > 0$, therefore, when

$$(N_{ij}^2 - \sum_k N_{ijk}^2)\alpha + (2N_{ij}^2 N_{ijm} - 2N_{ij} \sum_k N_{ijk}^2) < 0$$

holds, which means,

$$\alpha < \frac{2N_{ij}\sum\limits_k N_{ijk}^2 - 2N_{ij}^2 N_{ijm}}{N_{ij}^2 - \sum\limits_k N_{ijk}^2},$$

$D(\frac{N_{ijk}+\alpha\theta_{ijk}^*}{N_{ij}+\alpha}) - D(\frac{N_{ijk}}{N_{ij}}) < 0$. Therefore, in general, when the equivalent sample size α meets the following requirement

$$0 < \alpha < \frac{2N_{ij}\sum\limits_k N_{ijk}^2 - 2N_{ij}^2 N_{ijm}}{N_{ij}^2 - \sum\limits_k N_{ijk}^2},$$

the proposed Max-Min Fitness estimation is more efficient than Maximum Likelihood estimation.

□