Bayesian Network Learning and Inference
ECSE 6810 Fall Project 2

Wenting Li

December 19, 2018

1 Introduction

Given the datasets D, the distribution P underlying the D can be captured by a Bayes Network $\{G, \theta\}$. As both the structure G and the parameters θ need to be determined, the crucial points of Bayes learning include the structure learning and parameter learning. Due to the availability of the structure and the observed data, there are four types of Bayes learning. 1) "Parameter learning" means to learn the parameters when the structure G is known; 2) "incomplete parameter learning" means to learn the parameters if the data are partially observed and the structure G is known; 3) if the structure G is not known and the data are fully observed, then the learning is defined as "structure and parameter learning"; 4) the more challenging learning is the "incomplete structure and parameter learning" when the structure is unknown and the data are not fully observed.

The most basic learning issue is the "parameter learning" given the complete observations D. There are commonly two ways of generative learning methods: the maximum likelihood (ML) estimation and the Bayes estimation. The ML method is to maximize the log likelihood of $\log(p(D|\theta))$, and the Bayes estimation is to maximize the posterior log likelihood $\log(p(\theta|D))$, which actually is equivalent to be $\log(p(D|\theta)) + \log(p(\theta))$. Obviously, the difference between these two methods is that there is an extra prior distribution $\log(p(\theta))$ in the Bayes estimation method. It is found that when the size of datasets is small, the influence of prior distribution will influence the parameter learning results significantly, but if the datasets are large enough, the influence of the prior becomes trivial, thus both the ML and Bayes may obtain the similar results. In addition, if the application of the parameters are for the classification, the discriminative learning can be applied.

The "structure and parameter learning" is another fundamental learning to determine the topology of a Bayes Network when the observations are completely provided. To determine whether there is any links between two nodes, there are two main criteria. One is the score-based approach, and the other is the independent test-based method. For the independence test based method, the performance is not satisfying if the network is small or the datasets are not sufficient.

For the score-based method, the central idea is to define a score of the current structure G', search all possible way of changing the topology, and then check whether the score can be improved after the modification. The definition of the score is basically to maximize the marginal structure likelihood $p(D|G)$. Due to various assumptions, there are various score, like the Bayesian

Information Criterion (BIC), Bayesian Score, Akaike Information Criteriaon (AIC) are defined. Then these scores are applied on different scenarios to determine the optimal structure. One significant advantages of these score-based approach is that the score of the whole structure can be decomposable, thus the structure can be modified locally.

Among the proposed scores, the BIC score is extensively applied, as this score balances the model complexity and the closeness to the optimal solution and significantly improve the computation efficiency. The assumption of this score include that the prior distribution is uniform, the datasets are sufficient, moreover, the structure based on this score is usually local optimal, thus is influenced by different initial conditions. To implement the BIC score method, one feasible way is the Hill Climbing learning algorithm based on the Heuristic search. This method can locally modify the structure and check the BIC score afterward, the topology can only be changed if a larger BIC score can be obtained. This algorithm is efficient but can only ensure the local optimal.

Therefore, the aim of this project is to learn the structure based on the BIC score method given the training datasets. As one of the node representing the "target", which determines the type of the datasets, the type of each dataset can be determined by computing the maximum a posterior inference of the target node. Thus in the testing stage, the type of the datasets can be determined and the accuracy of testing will be obtained.

The remaining parts of the report are organized as follows: Section 2 describes and discusses the theory of Bayesian learning, inference methods. Then Section 3 validates the theory by learning the structure of the given training datasets, and the learned structures and classification accuracy illustrates the feasibility of the BIC score method. The Section 4 summarizes the tasks and concludes the results.

2 Theory and Analysis of Bayesian Learning and Inference

2.1 Bayesian Structure Learning Method

BN structure learning is to simultaneously learn the links among nodes and conditional probability for the nodes. It is more challenging than CN parameter learning.

There are two major types of Bayesian structure learning method: one is score-based method, and the other is independence-test based method. In this project we will implement one the the first type of method: the Hill Climbing learning method (HCL).

This method is a greedy method that iteratively update the local structure of each node based on the Bayesian Information Criterion (BIC) until convergence. Given the training datasets $D = \{D_1, \cdots, D_M\}$, and the BN structure G of N nodes $D^m = \{x_1^m, \cdots, x_N^m\}$, the derivation of BIC is to maximize the log structure likelihood $\log P(D|G)$. Notice that the parameters θ is unknown.

$$\begin{aligned} \log P(D|G) &= \Sigma_{m=1}^{M} P(D^m|G) && (1) \\ &= \Sigma_{m=1}^{M} \int P(D^m|\theta, G)P(\theta|G)d\theta && (2) \end{aligned}$$

As the iterm $\int P(D^m|\theta, G)P(\theta|G)d\theta$ is complicated to compute, it is assumed that the $P(D^m, \theta|G)$

follows a Gaussian distribution,

$$P(D^m,\theta|G) \sim \frac{1}{(2\pi|\Sigma|^{1/2})} e^{\frac{(\theta-\theta_0)^T\Sigma^{-1}(\theta-\theta_0)}{2}} \tag{3}$$

where Σ is the covariance matrix of θ.

then the Laplace Approximation method can estimate this item. According to the second-order Taylor expression, we expand the $\log P(D^m,\theta)$ at the mode θ_0,

$$\log(P(D^m,\theta|G)) \approx \log P(D^m,\theta_0|G) + (\theta-\theta_0)\frac{\partial \log P(D^m,\theta_0|G)}{\partial\theta} + (\theta-\theta_0)^T\frac{\partial^2 \log P(D^m,\theta_0|G)}{2\partial^2\theta}(\theta-\theta_0) \tag{4}$$

As the mode θ_0 satisfies that $\frac{\partial \log P(D^m,\theta|G)}{\partial\theta} = 0$, and take exponential function on both side of (4), we have

$$(P(D^m,\theta|G)) \approx P(D^m,\theta_0|G)e^{\overline{\frac{-1}{2(\theta-\theta_0)^T A(\theta-\theta_0)}}} \tag{5}$$

where $A_m = -\frac{\partial^2 \log P(D^m,\theta_0|G)}{2\partial^2\theta}$.

let

$$q(D^m,\theta) = P(D^m,\theta_0|G)e^{\overline{\frac{-1}{2(\theta-\theta_0)^T A_m(\theta-\theta_0)}}}$$

approximate $\log P(D^m,\theta|G)$.

$$\begin{aligned}
P(D|G) &= \Pi_{m=1}^M P(D^m|G) && (6)\\
&= \Pi_{m=1}^M \int P(D^m,\theta|G)d\theta && (7)\\
&\approx \Pi_{m=1}^M \int P(D^m,\theta_0|G)e^{\overline{\frac{-1}{2(\theta-\theta_0)^T A_m(\theta-\theta_0)}}}d\theta && (8)\\
&= \Pi_{m=1}^M P(D^m,\theta_0|G)(2\pi)^{d/2}|A_m|^{-\frac{1}{2}} && (9)\\
& && (10)
\end{aligned}$$

where d is the dimension of the θ.

$$\log P(D|G) = \Sigma_{m=1}^M \log P(D^m|\theta_0,G) + \log P(\theta_0|G) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\Sigma_{m=1}^M \log|A_m| \tag{11}$$

Assume there are a large number of datasets, M goes to ∞, then $\Sigma_{m=1}^M|A_m| = d\log M$, and assume the prior θ follows the uniform distribution, thus

$$\begin{aligned}
\theta^* &= \max_\theta \log P(D|G) && (12)\\
&\approx \max_\theta(\Sigma_{m=1}^M \log P(D^m|\theta_0,G) + \log P(\theta_0|G) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\Sigma_{m=1}^M|A_m|) && (13)\\
&\approx \max_\theta(\Sigma_{m=1}^M \log P(D^m|\theta_0,G) - \frac{d}{2}\log M && (14)
\end{aligned}$$

Therefore, the $S_{\text{BIC}}(G(X_n))$ is defined as

$$S_{\text{BIC}}(G) = \Sigma_{m=1}^{M} \log P(D^m|\bar{\theta}, G) - \frac{d(G)}{2} \log M \tag{15}$$

where $\bar{\theta}$ is the optimal parameters for the structure G, $d(G)$ is the number of independent parameters of the network G.

Applying the Bayes Chain rule to the log likelihood term of (15), the equation becomes

$$\begin{aligned} S_{\text{BIC}}(G) &= \Sigma_{m=1}^{M} \Sigma_{n=1}^{N} \log P(X_n^m|\bar{\theta}_n, G(X_n)) - \Sigma_{n=1}^{N} \frac{d(G(X_n))}{2} \log M && (16) \\ &= \Sigma_{n=1}^{N} \Sigma_{m=1}^{M} \log(P(X_n^m|\bar{\theta}_n, G(X_n)) - \frac{d(G(X_n))}{2} \log M && (17) \\ &= \Sigma_{n=1}^{N} S_{\text{BIC}}(G(X_n)) && (18) \end{aligned}$$

where $G(X_n)$ denotes the network for the nth network consisting the node X_n and its parents, $d(G(X_n))$ denotes the number of links in $G(X_n)$, $S_{\text{BIC}}(G(X_n))$ denotes the BIC score the network for the nth node, and $d(G(X_n))$ denotes the number of independent parameters, i.e., if the node has K states and J configurations, then $d(G(X_n)) = (K-1)J$.

Plug (6) into (1), then we have The procedures are summarized in the Algorithm 1.

Algorithm 1 The Hill Climbing learning Method

```
1: Input: the training datasets $D$ and the initialized BN structure $G$ and the initial BIC.
2: For the $i$th node ($i = 1, \cdots, N$), refine the structure of BN with the training data by one operation such as removing or adding or changing link direction of this node at one time, and obtain all possible new networks $G_j, j = 1, \cdots, i_n$.
3: Compute the maximum BIC score of the structure $G_{j^*}$ among all the $G_j$. If this new BIC is larger than the existing BIC, then save the structure $G = G_{j^*}$, otherwise keep the existing structure $G$.
4: Repeat the steps from 2-4 until furture changes of $G_{j^*}$ cannot improve the BIC score.
5: Output: $G$ and BIC.
```

Notice that in this project, the initialized structure G is given, and the operations are limited to deleting and reversing the directions.

2.2 Discussion of hill-climbing learning algorithm

This method is based on the BIC score. The BIC score is to formulated by three assumptions (1)($P(D^m, \theta|G)$) follows the Gaussian distribution, (2) the BN parameters $P(\theta|G)$ follows the uniform prior, (3) there are sufficiently large number of datasets that $M \longrightarrow \infty$. Thus if there are not enough datasets, or the parameters have some special prior rather than uniformly distributed, the BIC score is not a suitable criterion to establish the structure.

Moreover, this greedy algorithm can practically produce satisfactory structure efficiently, but the these results are usually influenced by different initializations. There is no theoretical guarantees of the optimal structure. Various techniques have been proposed to reach a better local optimal, such as Random Restart, TABU search, and Simulated Annealing method.

2.3 Bayesian Parameter Learning

There are three methods to learn the BN parameters. One is maximize the log likelihood (ML) without considering the prior distribution of θ, the second is the Bayesian BN learning, which assume the Dirichlet or Gaussian, or L1-norm prior, and the third method is the discriminative learning, which is better for classification problem. The first two methods are decomposable but the third one is not. Here The ML method is applied to estimate the parameters.

Let the BN parameter set $\theta = \{\theta_1, \cdots, \theta_N\}$ corresponding to the nodes. These parameters can be learned by the MLE method:

$$\theta^* = \arg\max_{\theta} P(\mathbb{D}|\theta) \tag{19}$$

$$= \arg\max_{\theta} \log(P(\mathbb{D}|\theta)) \tag{20}$$

$$= \arg\max_{\theta} LL(D:\theta) \tag{21}$$

$$= \arg\max_{\theta} \Sigma_{m=1}^{M} \log(P(D^m|\theta)) \tag{22}$$

$$= \arg\max_{\theta} \Sigma_{m=1}^{M} \Sigma_{n=1}^{N} \log(P(X_n^m|\pi(X_n^m))) \tag{23}$$

where $\pi(X_n^m)$ is the parents of X_n^m.

Therefore, we can learn each parameter θ_n independently.

$$\theta_n^* = \arg\max_{\theta} LL(D:\theta_n) \tag{24}$$

$$= \arg\max_{\theta} \Sigma_{m=1}^{M} \log(P(X_n^m|\pi(X_n^m))) \tag{25}$$

$$\tag{26}$$

For the discrete BN, the node $X_n \in \{1, 2, \cdots, K\}$ has K states. Assume there are J configurations of the parents of node X_n, and the parameter for the parameter of configuration j when node $X_n = k$ is θ_{njk}, and these configurations are independent, then we can further derive the simplify $P(X_n^m|\pi(X_n^m))$ for node X_n,

$$P(X_n^m|\pi(X_n^m)) = \Pi_{j=1}^{J} \theta_{nj}^{\mathbb{1}(\pi(X_n^m)=j)} \tag{27}$$

$$= \Pi_{j=1}^{J} \Pi_{k=1}^{K-1} \theta_{njk}^{\mathbb{1}(X_n^m=k,\pi(X_n^m)=j)} (1-\Sigma_{k=1}^{K-1}\theta_{njk})^{\mathbb{1}(X_n^m=K,\pi(X_n^m)=j)} \tag{28}$$

Plug (27) into (24), then we obtain

$$\theta_n^* = \arg\max_{\theta} \Sigma_{m=1}^{M} \log(\Pi_{j=1}^{J} \Pi_{k=1}^{K-1} \theta_{njk}^{\mathbb{1}(X_n^m=k,\pi(X_n^m)=j)} (1-\Sigma_{k=1}^{K-1}\theta_{njk})^{\mathbb{1}(X_n^m=K,\pi(X_n^m)=j)}) \tag{29}$$

$$= \arg\max_{\theta} \Sigma_{m=1}^{M} \Sigma_{j=1}^{J} \Sigma_{k=1}^{K-1} \theta_{njk} \mathbb{1}(X_n^m = k, \pi(X_n^m) = j) \log(\theta_{njk}) \tag{30}$$

$$+ \mathbb{1}(X_n^m = K, \pi(X_n^m) = j) \log(1 - \Sigma_{k=1}^{K-1}\theta_{njk})) \tag{31}$$

$$= \arg\max_{\theta} N_{njk} \log(\theta_{njk}) + N_{njK} \log(1 - \Sigma_{k=1}^{K-1}\theta_{njk})) \tag{32}$$

where N_{njk} is the total number of node $X_n = k, \pi(X_n^m = j)$ of the training data.

According to the first order optimal condition, θ_n^* satisfy that $\frac{\partial(LL(D:\theta_n))}{\partial\theta_n} = 0$, then

$$\theta_n^* = \frac{N_{njk}}{\Sigma_{l=1}^K N_{njl}} \tag{33}$$

2.4 Discussion of MLE parameter leanring

This project is a classification problem, the discriminative learning may have better performance, but as required by ML method, here we only show the results of ML method. Since ML method is decomposable, the implementation is much easier.

2.5 Bayesian MAP inference for classification

When the 10 features are given (like the test dataset 1) and the first node is the target, the classification issue becomes to deterministic and is $P(X_1|X_2,\cdots,X_{11})$ directed from the table CPT of the node 1. Let the true label of the ith target be x_i while the predicted by the MAP inference be $\bar{x}_1$.

$$\bar{x}_1 = \arg\max_{X_1} P(X_1|X_2,\cdots,X_{11}) \tag{34}$$

Thus the classification accuracy is defined by $\eta_i, i = 1,2$, where η_i denotes the classification accuracy for the Test data i.

$$\eta = \frac{\text{The number of corrected classified}}{\text{Total test samples}}$$

where “The number of corrected classified” denotes the number of testing datasets that $x_i = \bar{x}_i$.

When only partial features are given, the Gibbs sampling method is applied to estimate the states of the unknown features and then obtain the MAP inference of the target. For the dataset 2, the classification is equivalent to obtain the inference

$$\bar{x}_1 = \arg\max_{X_1} P(X_1|X_2,\cdots,X_6) \tag{35}$$

Although there are many methods can be employed to implement the inference, include logic sampling, weighted logic sampling , belief propagation and variable elimination, here the Gibbs sampling is selected due to the accuracy guarantees.

The Gibbs sampling

Notice that the transition model $p(X_j|\mathbf{x}_{-j}^i,\mathbf{e})$ satisfies

$$p(X_j|x_{-j}^i,\mathbf{e}) = p(X_j|MB(X_j)) \tag{36}$$

$$= \frac{p(X_j|\pi(X_j))\Pi_{i=1}^k p(Y_i|\pi(Y_i))}{\Sigma_{x_j} p(x_j|\pi(x_j))\Pi_{i=1}^k p(Y_i|\pi(Y_i))} \tag{37}$$

where $\pi(X)$ denotes the parents of X, Y_i is the ith child of X_j’s kth children.

Algorithm 2 The Single Chain Gibbs Sampling Method

```
Input: burn in period $t$, skip steps $k$, and iteration times $N$.
Initialize all the unknown nodes $\mathbb{X} = \{X_1, X_7, \cdots, X_{11}\}$ by some random binary numbers according to their total number of states, and known evidence are $\mathbf{e} = X_2, \cdots, X_6$. Let $n = 0$;
for $i = 1, \cdots, N$ do
    Random pick the $j$th state of the unknown states;
    Update the value of the state $X_j$ by obtaining the sample $x_j^{i+1}$ following the distribution $p(X_j|x_{-j}^i, \mathbf{e}) = p(X_j|MB(X_j))$ of (36);
    while other states are kept the same $x_k^{i+1} = x_k^i, k \neq j$;
    Form a sample of $\mathbf{x}^{i+1}$;
    if $i = t + nk$ then
        Return the sample $\mathbf{x}^{t+nk}$                    ▷ Collecting Sampling Results
        $n = n + 1$;
Output: $\bar{x^i}_1 = \arg\max_{X_1} P(X_1|X_2, \cdots, X_6)$
```

3 Experimental Results

3.1 Datasets

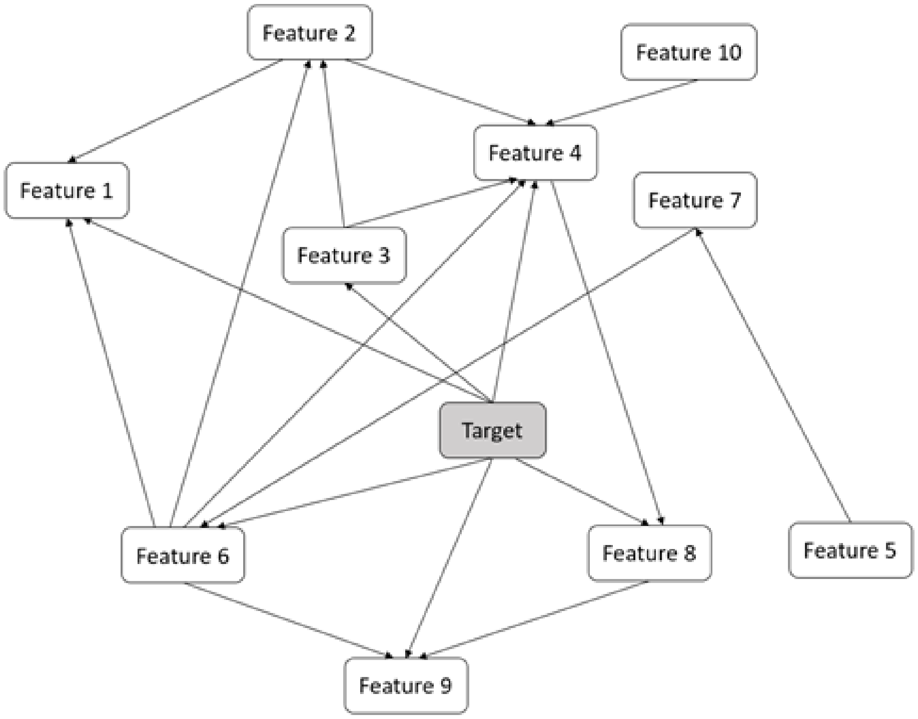

Fig. 1: *The initialized BN*

Database of baseball players and play statistics. Features are statistics of baseball players from different aspects and they have been quantified into binary values 0 or 1. Target is the voting for hall of fame. It has three values: 0, 1, and 2.

Variables Target (“Hall of fame”):0,1,2
10 features are binary states (0 or 1)
Feature 1: “number of seasons” = 0, 1
Feature 2: “at bats” = 0, 1
Feature 3: “runs” = 0, 1
Feature 4: “triples” = 0, 1
Feature 5: “Home runs” = 0, 1
Feature 6: “RBIs”=0, 1
Feature 7: “strikeouts” = 0, 1
Feature 8: “Batting average” = 0, 1
Feature 9: “slugging pct” = 0, 1
Feature 10: “fielding average” = 0, 1
The data consists of 900 training samples, 150 samples for testing data 1 and another 150 samples for testing data 2. The data and the readme file can be obtained from the link below.
https://www.dropbox.com/sh/9kpz7xwwdehmium/AADauJgL-tY3n7-83ZgbmLj4a?dl=0

3.2 Tasks

Perform the following tasks:

- Implement the hill climbing method to learn the BN structure. Display the learnt BN structure

- Implement the ML method to learn the BN parameters

- Perform MAP inference to determine the most likely class value for the target node for the given testing data. The testing data consists of two sets: test data 1 and test data 2. Test data 1 contains 150 samples with complete feature values for each sample. Test data 2 contains another 150 samples with feature values for only the first five features. Show the average classification accuracy for each class for both testing datasets.

3.3 The BN Structure

The process of changing the structure is tracked as follows:
The BIC is -3070.2507 After delete link 2 of the node 1
The BIC is -2990.4065 After delete link 5 of the node 1
The BIC is -2934.5514 After reverse link 5 of the node 1
The BIC is -2918.8662 After delete link 1 of the node 2
The BIC is -2899.9634 After reverse link 1 of the node 2
The BIC is -2888.1257 After delete link 5 of the node 3
The BIC is -2886.8475 After reverse link 5 of the node 3
The BIC is -2869.4763 After reverse link 3 of the node 4

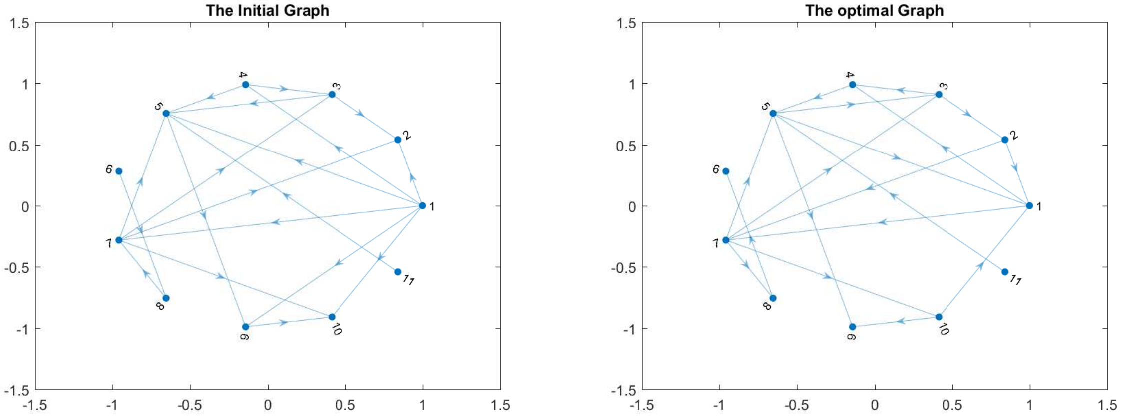

Fig. 2: The initialized and the optimal BN Graph

```
The BIC is -2864.6991 After reverse link 11 of the node 5
The BIC is -2864.6991 After reverse link 8 of the node 6
The BIC is -2854.9109 After reverse link 8 of the node 7
The BIC is -2850.9231 After reverse link 10 of the node 9
The BIC is -2848.3038 After delete link 1 of the node 10
The BIC is -2839.6894 After reverse link 1 of the node 10
The BIC is -2839.6894 After 1 iterations
The BIC is -2839.3193 After delete link 2 of the node 1
The BIC is -2835.5462 After delete link 9 of the node 1
The BIC is -2815.1991 After reverse link 7 of the node 2
The BIC is -2815.1991
```

3.4 The final and inital BIC scores

Table 1: *The BIC scores*

State	S_{BIC}
Initial	-3085.9
Final	-2815.2

3.5 Classification accuracy

To classify the second datasets, the Gibbs Sampling method is applied. The sample number is set to be 5000, the burn-in period is 1000, the skip time is 100 and the single Markov Chain is employed to generate samples. In Table 2, $\eta_i, i = 1,2$ denotes the classication accuracy for the ith testing datasets.

Table 2: *The Classification Accuracy*

State	η_1 %	η_2 %
Initial Graph	89.33	91.33
Final Graph	89.33	90.67

Discussion of the Results The results of $\eta_i, i = 1,2$ are interesting since the optimal graph does not increase the classification accuracy. When looking into the reasons, it is found that the parameters learned by maximizing the log likelihood $\log P(D|\theta)$ are not optimal for classification.

$$\begin{aligned} \theta^* &= \arg\max_\theta \Sigma_{m=1}^{M} \log(P(D^m|\theta)) && (38) \\ &= \arg\max_\theta \Sigma_{m=1}^{M} \log(P(X_1^m, \cdots, X_n^m|\theta)) && (39) \end{aligned}$$

The θ^* is learned to maximize the joint probability but what we really care about classification is the conditional probability $P(X_1^m|X_2^m, \cdots, X_n^m|\theta)$, where X_1^m is the target for the mth dataset.

To reach a better classification performance, the optimal parameters θ should be learned by maximizing the conditional likelihood instead of the joint likelihood of (38). Thus for classification problem, the optimal parameters for classification should be learned by

$$\theta^* = \arg\max_\theta \Sigma_{m=1}^{M} \log(P(X_1^m|X_2^m, \cdots, X_n^m, \theta)) \tag{40}$$

The issue of computing the parameters based on (40) is that this equation is not decomposable in terms of n nodes, thus the computational complexity increases when the nodes are many.

3.6 The BN Structure with another initialization

The new initialization is shown in the left graph of the Fig. 3. We can observe that the optimal graph on the right of Fig. 3 is different from that of Fig. 2. The difference between the initial graph in Fig. 3 with that Fig. 2 is that more connections are added to the initial graph. The intuition why this initial graph may obtain a higher BIC is that in this project only deletion and reverse operations are allowed, thus there is no possibility to add more links, therefore, if there are more links in the initial graph, there are more possibility to search a better graph.

3.7 The BIC scores accuracy with another initialization

When try a different initialization, the optimal graph can be various as shown in Fig. 3, and the BIC score shown in Table 3 can be improved from -2801.8 to -2736.3. The classification is also computed and the η_1 = 90.67%, which is higher than the 89.33% in the Table 1 with the original initialization.

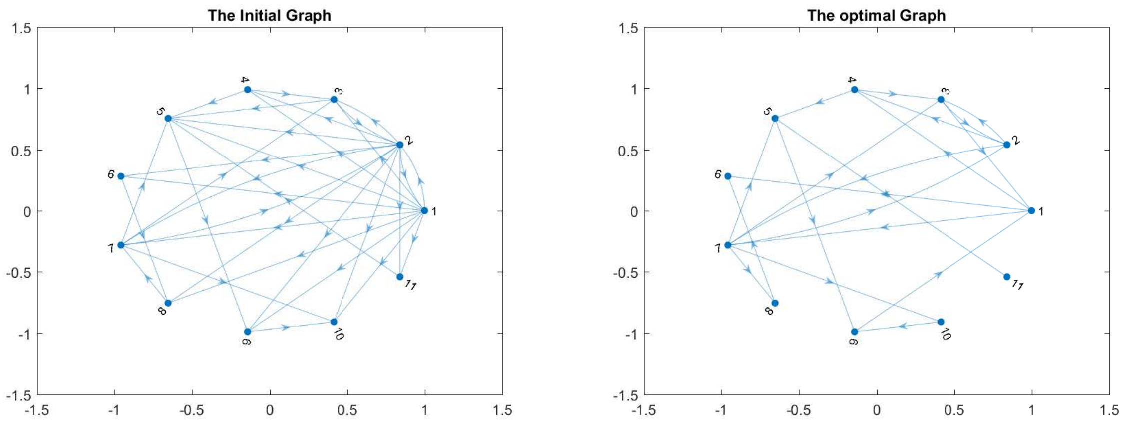

Fig. 3: *The initialized and the optimal BN Graph with another initialization*

Table 3: *The BIC scores with another initialization*

State	S_{BIC}
Initial	-3247.1
Final	-2736.3

Table 4: *The Classification Accuracy*

State	η_1 %	η_2 %
Initial Graph	89.33	91.33
Final Graph	90.67	92.00

3.8 Classification accuracy with another initialization

The classification results are shown in Table 4. The η_i of the Final Graph has a better classification accuracy rate than that of the Initial Graph for the first and the second testing datasets. The $\eta_1 = 90.67$ of the final graph is better than that of the final graph with the original initialization in Table 2, but the η_2 in Table 4 is no better than that in Table 2. Thus a high BIC score cannot ensure the classification accuracy rate is always better on the testing datasets.

4 Conclusions

1. A summary of the tasks:
    - The BN structures are learned by the hill-climbing algorithm;
    - The parameters of the BN are learned by maximizing likelihood;

- The two testing datasets are classified by the MAP inference method after approximating the posterior probability through Gibbs sampling method;
- All the above tasks are performed on another BN structure learned by a different initialization. This different BN structure achieves a higher BIC score.
- The theoretical analysis of the structure and parameter learning methods are described, and the experimental results are analyzed.

2. The pluses and minuses of BN classifiers v.s. deterministic classifiers:

   - The pluses of BN classifiers include 1) the BN classifiers output the probability of each class instead of the type simply. The probability provides the confidence of the classification results. For example, with probability of 99% being the "class 1" has more confidence than with probability of 51% being the "class 1", although the final output types for these two conditions are both "class 1". 2) BN classifiers give the user the opportunites to add the prior knowledge to the classifiers rather than simply relying on the data. The prior knowledge is significant especially when the training datasets have a small size.
   - The minuses of BN classifiers: 1) the main issue is that the structure learnt by BN method can achieve a high BIC score or maximum likelihood score but may have a poor classification performance. Thus the classification performance of BN may not be better than the deterministic method. 2) When more nodes are included into the BN, the complexity of modeling and computing inference is increased significantly, which rises up the computation and storage cost.

3. Issues discovered: (1) although the hill-climbing learning can ensure that the final structure to achieve a high BIC score, the performance of classification of the testing datasets is not guaranteed to be optimal.

   For example, if only allow changing the links of each node's parents instead of all the links, another graph G' with the local optimal BIC -2806 is obtained, which is no better than the optimal solution in the Table 1, but the $\eta_2 = 92.67$ using the G', which is higher than the 90.67 in Table 2. Therefore, a high BIC score cannot ensure a high classification accuracy rate.

4. Learned: Although the theory of BIC is simple, there are many conditions need to be considered when implementing this method for the practical issue. For example, whether the training datasets are large enough, whether the changing of the structure is under the graph constraint, and what if there are no samples in the training datasets corresponding to certain configurations, thus more boundary or abnormal detection should be considered.