Project3: BN Structure Learning under Incomplete Data

1. Introduction

Bayesian Network (BN) learning problem is about learning a BN that characterizes the independence relationship between variables within a dataset. Since describing a probabilistic graph model (PGM) requires knowledge of its structure and parameters, the learning problem consists of learning the network structure, called structure learning, and parameters, parameter learning. In previous homework, we extensively discussed BN parameter learning, and in this project, we will focus on structure learning.

Compared to parameter learning, structure learning is a more general learning method because it does not make assumptions about network structure $\mathcal{G}$. The only prerequisite of BN structure learning is the knowledge of nodes $X = \{x_1, ..., x_i, ..., x_N\}$ and data $\mathcal{D} = \{D_1, ..., D_i, ..., D_M\}$ (either complete or incomplete). In some sense, structure learning is an outer loop for parameter learning. The structure $\mathcal{G}$ of BN decides the upper bound of the likelihood its parameters can reach.

Like any other learning problem, bias-variance tradeoff playing an important role in structure learning. Naively adding links between nodes results in a higher likelihood bound while meanwhile putting the learned PGM at the risk of overfitting. A common heuristic control strategy is introducing a regularizer to balance the model's complexity and the likelihood bound, which forms the idea of score-based structure learning.

In this project, we implemented the score-based structure learning on a toy problem. We first reviewed the current progress about structure learning and examined the performance of each method.

2. Score-based learning

As we mentioned in the introduction section, score-based learning's object is to balance the likelihood bound and model complexity. The mathematical formulation of its object in structure learning is called score function. One popular score function is called *Bayesian Information Criterion* score (BIC).

2.1 Bayesian information criterion

Using BIC as score function falls into the category of maximum likelihood learning. The mathematical formulation of this problem can be written as:

$$\arg\max_{\mathcal{G}} \log p(\mathcal{D}|\mathcal{G}) \tag{2.1}$$

Equation 2.1 is intractable and not implementable. We need to further expand it as:

$$\begin{aligned} &\max_{\mathcal{G}} \log p(\mathcal{D}|\mathcal{G}) \\ =&\max_{\mathcal{G}} \log \int_{\theta} p(\mathcal{D}, \theta|\mathcal{G}) d\theta \end{aligned} \tag{2.2}$$

Still, equation 2.2 is not computable due to the integration over θ but allows approximation. One approximation technique is Laplace approximation (Ji's 2020 PGM BN learning lecture note, page 53, see more detail at appendix 7.1):

$$\begin{aligned} &\max_{\mathcal{G}} \log \int_{\theta} p(\mathcal{D}, \theta|\mathcal{G}) d\theta \\ \approx &\max_{\mathcal{G}} \log p(\mathcal{D}|\theta^{G}_{\text{MAP}}, \mathcal{G}) + \log p(\theta^{G}_{\text{MAP}}|\mathcal{G}) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|A| \end{aligned} \tag{2.3}$$

Where d is the degree of freedom of θ and A is the negative Hessian matrix. When the size of data is large $M \to \infty$, $\log|A| \approx d \log M$ (Ji, 2019, page 80). See appendix 7.2 for more detail.

Equation 2.3 is the core equation for structure learning. Based on the choice of the second term (parameter's prior) of equation 2.3, the structure learning's score function may vary. If the parameter prior is a non-informative Dirichlet distribution

(uniform distribution), the score function is called the Bayesian Information Criterion (BIC).

With all these assumptions, the structure learning object using BIC score function can be written as:

$$\begin{aligned} &\max_{\mathcal{G}} S_{BIC}(\mathcal{G}) \\ =&\max_{\mathcal{G}} \log p(\mathcal{D}|\theta^G_{\text{MAP}}, \mathcal{G}) - \frac{d(\mathcal{G})\log M}{2} \end{aligned} \tag{2.4}$$

Equation 2.4 is the equation for the whole Bayesian network model. One basic property of the Bayesian network is its joint distribution is factorizable. As a result of this factorization, the likelihood of a well-parameterized Bayesian network is decomposable (Ji, 2019, equation 3.95):

$$\begin{aligned} &\max_{\mathcal{G}} S_{BIC}(\mathcal{G}) \\ =&\max_{\mathcal{G}} \log p(\mathcal{D}|\theta^G_{\text{MAP}}, \mathcal{G}) - \frac{d(\mathcal{G})\log M}{2} \\ =&\max_{\mathcal{G}_1,\mathcal{G}_2,\ldots,\mathcal{G}_N} \sum_{i=1}^{N} \log p(\mathcal{D}|\theta^G_{\text{MAP}}, \mathcal{G}_i) - \sum_{i=1}^{N} \frac{d(\mathcal{G}_i)\log M}{2} \\ =&\max_{\mathcal{G}_1,\mathcal{G}_2,\ldots,\mathcal{G}_N} \sum_{i=1}^{N} [\log p(\mathcal{D}|\theta^G_{\text{MAP}}, \mathcal{G}_i) - \frac{d(\mathcal{G}_i)\log M}{2}] \\ =&\max_{\mathcal{G}_1,\mathcal{G}_2,\ldots,\mathcal{G}_N} \sum_{i=1}^{N} S_{BIC}(\mathcal{G}_i) \end{aligned} \tag{2.5}$$

Where $\mathcal{G}_i = \{X_i, \pi(X_i)\}$ is a local structure of the graph that contains a node Xi and its parents $\pi(X_i)$.

2.2 Hill climbing

Once we defined the score function, we can see the structure learning as an optimization problem, which can be solved by searching over the structure and parameter space. However, this combinatorial searching problem is NP-hard if we

do not restrict the BN structure to a specific structural set (Ji, 2019, page 82). As a consequence, we need to resort to some heuristic strategies to approximate the local solution.

In project, we are instructed to implement hill-climbing algorithm (Chikering, 2003) to learn BN structure. The core idea of hill-climbing is to

- *enumerate* some possible structure-parameter combinations,
- *evaluate* each combination, and
- *select* the top severals instances as the solution.

The enumerating operation including three operations: adding links, removing links, and reversing link directions. The evaluation step is to compute the maximum likelihood the structure allows. The select step includes the greedy method (Chikering, 2003), choose and save the top1 candidate, and simulated annealing (Ji, 2019, page 83), weighted and sum over the top N candidates. Despite bringing more computation burdens, simulated annealing is usually a preferred solution due to its robustness in the nonlinear problem, including the BN structure learning problem.

Algorithm 1 shows the pseudo code of hill climbing algorithm (adopted and modified from (Ji, 2019, page 84, algorithm 3.7))

Algorithm 1 Likelihood weighted sampling

Init A Bayesian Network with a initial structure $\mathcal{G}^0$
Init A random order of the nodes $\{X_1, X_2, ..., X_N\}$
while not converging(BIC score still changes) **do**
    for $i = 1$ to N **do**
        Explore the structure in the following steps
        Remove links between X_i and the set of nodes excludes $\mathrm{X}\backslash X_i$
        Reverse links between X_i and the set of nodes excludes $\mathrm{X}\backslash X_i$
        Add links between X_i and the set of nodes excludes $\mathrm{X}\backslash X_i$
    end for
end while

2.3 Structure EM.

In simple, the structure learning is merely concatenating structure search and parameter learning, of which the protocol is the likelihood estimation in any of the score function including BIC. The same algorithm used in parameter learning can be used in structure learning with minor modifications.

Like other structure learning algorithms, structure EM structure combines structure search and EM parameter learning, and it can solve structure learning with incomplete data $\mathcal{Z}$. To use the EM algorithm, we introduced some inferred data. Thus we need to modify equation 2.2 to allow the implementation of the EM algorithm. Note that we denote the inferred data as $\mathcal{Z}$ and observed data as $\mathcal{Y}$.

$$
\begin{aligned}
&\max_{\mathcal{G}} \log p(\mathcal{D}|\mathcal{G}) \\
=&\max_{\mathcal{G}} \log p(\mathcal{Y}|\mathcal{G}) \\
=&\max_{\mathcal{G}} \log \sum_{\mathcal{Z}} p(\mathcal{Z},\mathcal{Y}|\mathcal{G}) \\
&\text{See appendix 7.3 for the derivation for this step} \\
=&\max_{\mathcal{G}} \mathbb{E}_q[\log p(\mathcal{Z},\mathcal{Y}|\mathcal{G})] - \sum_{\mathcal{Z}} q(\mathcal{Z}|\mathcal{Y},\Phi)\log q(\mathcal{Z}|\mathcal{Y},\Phi) \\
\Rightarrow&\max_{\mathcal{G}} \mathbb{E}_q[\log p(\mathcal{Z},\mathcal{Y}|\mathcal{G})]
\end{aligned}
\tag{2.5}
$$

Follow the same Laplace approximation method as we mentioned in section 2.1, we further expand the equation 2.5:

$$
\begin{aligned}
&\mathbb{E}_q[\log p(\mathcal{Z},\mathcal{Y}|\mathcal{G})] \\
=&\mathbb{E}_q[\int_{\theta} \log p(\mathcal{Z},\mathcal{Y}|\mathcal{G},\theta)d\theta] \\
&\text{Apply Laplace approximation} \\
=&\mathbb{E}_q[\log p(\mathcal{Z},\mathcal{Y}|\mathcal{G},\theta) - \frac{d(\mathcal{G})}{2}\log M]
\end{aligned}
$$

Algorithm 2 shows the pseudo code of structure EM algorithm (adopted and modified from (Ji, 2019, page 94, algorithm 3.11)):

Algorithm 2 The structural EM

Init A Bayesian Network with a initial structure $\mathcal{G}^0$
Init A random initialization of parameter Θ^0
t = 0
while not converging **do**
    E-step:
    for $m = 1$ to M **do**
        if x^m contains missing variables z^m **then**
            for $j = 1$ to $K^{|z^m|}$ **do**
                $w_{m,j} = p(\mathrm{z}_j^m | \mathrm{y}^m, \theta^t)$
            end for
        end if
    end for
    M-step:
    $\mathbb{E}_q(BIC(\mathcal{G})) = \sum_{m=1}^{M} \sum_{j=1}^{K^{|z^m|}} w_{m,j} \log p(\mathrm{y}^m, \mathrm{z}_j^m | \theta^t, \mathcal{G}) - \frac{d(\mathcal{G})}{2} \log M$
    $\mathcal{G}^{t+1}, \theta^{t+1} = \arg\max_{\mathcal{G}} \mathbb{E}_q(BIC(\mathcal{G}))$
    // Find $\mathcal{G}^{t+1}$ to maximize the expected BIC score through a local search algorithm, like hill-climbing method
    t = t+1
end while

3. Experiment Description.

In this project, we were instructed to learn the parameter and the structure given a set of incomplete data. Meanwhile, we were informed that there are five binary nodes: “A”, “B”, “C”, “D”, “E”.

Given this information, we need to learn a Bayesian Net, both the structure and the parameters.

4. Experiment Results:

Since the textbook does not explicitly introduce how to choose the convergence criterion, I make a guess and choose the following criteria:

- The structure of the BN does not change anymore.
- The parameters of the BN do not change anymore.
- The BIC of the BN do not change anymore.

My code satisfies the third criteria, but does not meanwhile satisfy the first and second criterion. Simply put, the structure I learned is hesitation two alternative structures. Here I picked and reported on possible learning results.

4.1 The learned structure and conditional probability table (CPT)

Figure1 shows the structure of the learned Bayesian network.

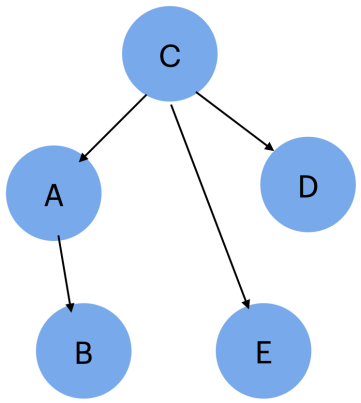

Figure1 Learned structure for the data

Meanwhile, I also learned another structure that returns similar BIC (see appendix 7.4). These two structures reached the highest BIC alternatively, but their difference are subtle ($\Delta < 1e-3$).

Table 1 shows the learned CPT.

Table1 learned CPT using EM with uniform initialization

Node			**CPT**
Node A			
		A=1	A=2
	C=1	0.693566	0.306434
	C=2	0.214970	0.785030
Node B			
		B=1	B=2

	A=1	0.275119	0.724881
	A=2	0.880999	0.119001
Node C			
		C=1	C=2
	prior	0.375129	0.624871
Node D			
		D=1	D=2
	C=1	0.287089	0.712911
	C=2	0.826710	0.173290
Node E			
		E=1	E=2
	C=1	0.021395	0.978605
	C=2	0.238280	0.761720

4.2 The BIC and log-likelihood as a function of iteration.

Figure2 shows the BIC score and log-likelihood score as a function of iterations. We may find that the structure EM converges around 8 iterations. The change in the following iterations on both BIC and log-likelihood scores are subtle.

Another observation is the difference between the BIC and log-likelihood increases along with the iteration. One interpretation of this observation is that at the beginning of the training, the structure was simple, the penalty term of model complexity $\frac{d}{2}\log M$ was small. As iteration lasts, the model becomes more and more complicated, so the penalty term increases. When the algorithm decided the structure, the penalty term was then a constant, and the difference between BIC and log-likelihood fix.

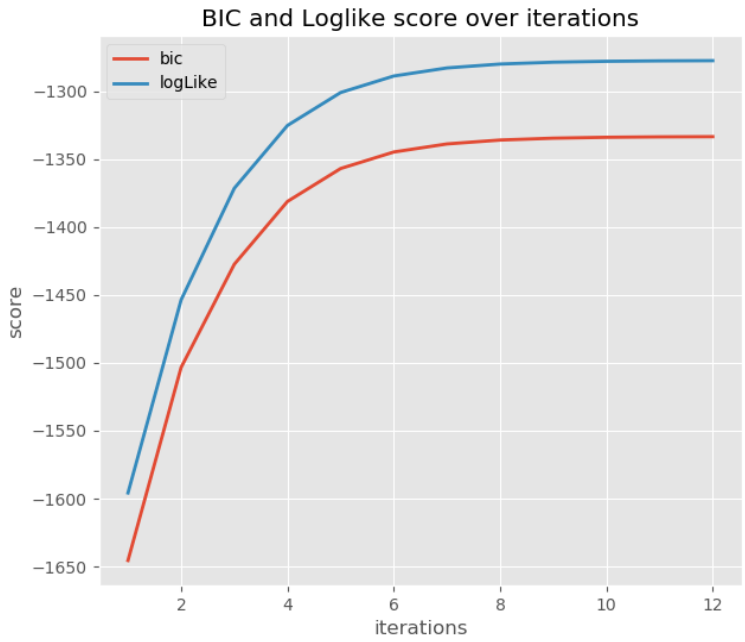

Figure2 BIC and logLikelihood score over iterations

5.Conclusion and discussion

It amazing to find that the structure can obtain such a good convergence property when using such a simple local search algorithm. Another interesting finding is its close relationship to Meta-Learning (MAML algorithm, Finn 2017). In my perspective, this algorithm replace the hill-climbing search when exploiting the structure of the model.

6. Reference

1. Ji, Q. (2019). Probabilistic Graphical Models for Computer Vision. Academic Press.
2. Chickering, D. M. (2002). Optimal structure identification with greedy search. Journal of machine learning research, 3(Nov), 507–554.
3. Ji, Q (2020). PGM lecture note, parameter learning.

4. Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. arXiv preprint arXiv:1703.03400.

7. Appendix

7.1 Laplace approximation

The following derivation follows the PGM parameter learning V4, page 53. The idea of Laplace approximation is using Gaussian distribution to approximate an arbitrary distribution. The simplest approximation of a distribution is to capture its first moment–expectation, which is also $\theta_{\text{MAP}} = \arg\max_\theta p(\theta)$. The Laplace approximation is find a Gaussian distribution of which the expectation match θ_{MAP} of the target distribution.

Here $\theta^G_{\text{MAP}} = \arg\max_\theta p(\mathcal{D}, \theta|\mathcal{G})$, and the Laplace approximation of $p(\mathcal{D}, \theta|\mathcal{G})$ can be written as (Ji, 2020, appendix 3.9.3):

$$\begin{aligned} &\log p(\mathcal{D}, \theta|\mathcal{G}) \\ \approx &\log p(\mathcal{D}, \theta^G_{\text{MAP}}|\mathcal{G}) + (\theta - \theta^G_{\text{MAP}})^\top \frac{\partial \log p(\theta^G_{\text{MAP}})}{\partial \theta} \\ &+ \frac{1}{2}(\theta - \theta^G_{\text{MAP}})^\top \frac{\partial^2 \log p(\theta^G_{\text{MAP}})}{\partial^2 \theta}(\theta - \theta^G_{\text{MAP}}) \end{aligned} \tag{7.1}$$

When $\theta = \theta^G_{\text{MAP}}$, the distribution reach the maximum. Thus, the first order Tyler's expansion equals to 0. Equation 7.1 can be written as:

$$\begin{aligned} &\log p(\mathcal{D}, \theta|\mathcal{G}) \\ \approx &\log p(\mathcal{D}, \theta^G_{\text{MAP}}|\mathcal{G}) + \frac{1}{2}(\theta - \theta^G_{\text{MAP}})^\top \frac{\partial^2 \log p(\theta^G_{\text{MAP}})}{\partial^2 \theta}(\theta - \theta^G_{\text{MAP}}) \end{aligned} \tag{7.2}$$

7.2 Derivation of BIC score function

With the Laplace approximation, we can further expand equation 2.2 as

$$
\begin{aligned}
&\max_{\mathcal{G}} \log \int_{\theta} p(\mathcal{D}, \theta | \mathcal{G}) d\theta \\
\approx &\max_{\mathcal{G}} \log \int_{\theta} p(\mathcal{D}, \theta^G_{\text{MAP}} | \mathcal{G}) \exp \frac{1}{2} (\theta - \theta^G_{\text{MAP}})^{\top} \frac{\partial^2 \log p(\theta^G_{\text{MAP}})}{\partial^2 \theta} (\theta - \theta^G_{\text{MAP}}) d\theta \\
= &\max_{\mathcal{G}} \log \int_{\theta} p(\mathcal{D}, \theta^G_{\text{MAP}} | \mathcal{G}) \exp -\frac{1}{2} (\theta - \theta^G_{\text{MAP}})^{\top} A (\theta - \theta^G_{\text{MAP}}) d\theta \\
= &\max_{\mathcal{G}} \log p(\mathcal{D}, \theta^G_{\text{MAP}} | \mathcal{G}) \int_{\theta} \exp -\frac{1}{2} (\theta - \theta^G_{\text{MAP}})^{\top} A (\theta - \theta^G_{\text{MAP}}) d\theta \\
= &\max_{\mathcal{G}} \log p(\mathcal{D}, \theta^G_{\text{MAP}} | \mathcal{G}) + \frac{d}{2} \log 2\pi - \frac{1}{2} \log |A| \\
= &\max_{\mathcal{G}} \log p(\mathcal{D} | \theta^G_{\text{MAP}}, \mathcal{G}) + \log p(\theta^G_{\text{MAP}} | \mathcal{G}) + \frac{d}{2} \log 2\pi - \frac{1}{2} \log |A|
\end{aligned}
\tag{7.3}
$$

7.3 Derivation of BIC with inferred data

With the Laplace approximation, we can further expand equation 2.5 as:

$$
\begin{aligned}
&\log \sum_{\mathcal{Z}} p(\mathcal{Z}, \mathcal{Y} | \mathcal{G}) \\
= &\log \sum_{\mathcal{Z}} q(\mathcal{Z} | \mathcal{Y}, \Phi) \frac{p(\mathcal{Z}, \mathcal{Y} | \mathcal{G})}{q(\mathcal{Z} | \mathcal{Y}, \Phi)} \\
\geq &\sum_{\mathcal{Z}} q(\mathcal{Z} | \mathcal{Y}, \Phi) \log \frac{p(\mathcal{Z}, \mathcal{Y} | \mathcal{G})}{q(\mathcal{Z} | \mathcal{Y}, \Phi)} \quad \text{Jensen's inequality} \\
= &\sum_{\mathcal{Z}} q(\mathcal{Z} | \mathcal{Y}, \Phi) \log p(\mathcal{Z}, \mathcal{Y} | \mathcal{G}) - \sum_{\mathcal{Z}} q(\mathcal{Z} | \mathcal{Y}, \Phi) \log q(\mathcal{Z} | \mathcal{Y}, \Phi) \\
= &\mathbb{E}_q [\log p(\mathcal{Z}, \mathcal{Y} | \mathcal{G})] - \sum_{\mathcal{Z}} q(\mathcal{Z} | \mathcal{Y}, \Phi) \log q(\mathcal{Z} | \mathcal{Y}, \Phi)
\end{aligned}
\tag{7.4}
$$

7.4 Alternative structure

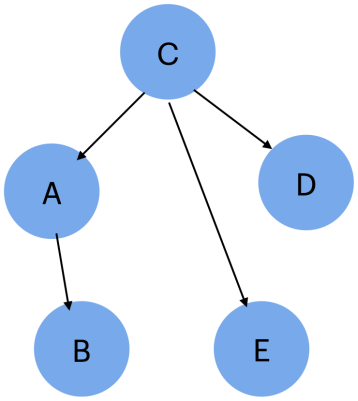

Figure7.1 Alternative structure for the data