Journal of Computational Statistics and Modeling, Vol. 1, No. 2, 87-97, December 2021

A Simple Gibbs Sampler for Learning Bayesian Network Structure

Vahid Rezaei Tabar*[1]

[1] Assistant Professor, Department of Statistics, Allameh Tabataba'i University, Tehran, Iran.

School of Biological Sciences, IPM, Tehran, Iran.

Recieved: 27/09/2020 Accepted: 28/08/2021

Abstract:

The aim of this paper is to learn the Bayesian network structure for discrete variables. For this purpose, we introduce a Gibbs sampler method. Each sample represents a Bayesian network. Thus, in the process of Gibbs sampling, we obtain a set of Bayesian networks. For achieving a single graph that represents the best graph fitted on data, we use the mode of burn-in graphs. This means that the most frequent edges of burn-in graphs are considered to indicate the best single graph. The results on the well-known Bayesian networks show that our method has higher accuracy in learning Bayesian network structure..

Keywords: Bayesian Network, Gibbs sampling, Burn-in graphs

Mathematics Subject Classification (2010): 99X99, 99X99.

*Corresponding Author: vhrezaei@atu.ac.ir

1. Introduction

A graphical model is a statistical model embodying a set of conditional independence relationships. We consider the structure learning for graphical models based on the directed acyclic graphs (DAGs), which are known as Bayesian network (BN) (Pearl , 1998). In BNs, the nodes are random variables, and the arcs specify the conditional independence structure between the random variables (Heckerman , 1998). The learning task in a BN can be separated into two subtasks, structure learning; which is to identify the topology of the network, and parameter learning; which is, to estimate the parameters (conditional probabilities) for a given network topology(Arias et al. , 2015).

One hypothetically well-founded approach for learning BN is to use the Markov Chain Monte Carlo (MCMC) techniques. This approach is very popular, and variations have been used by Madigan et al. (1995) and Giudici and Giudici (2003). Madigan et al. (1995) proposed the original version of the MCMC in which each move in the Markov chain consists of basically a single edge changes to the current graph (G). This algorithm is a classical Metropolis-Hastings sampler. The acceptance probability is $\min(1, r(G', G))$:

$$r(G', G) = \min\{1, \frac{\#nbd(G)P(G'|D)}{\#nbd(G')P(G|D)}\}, \tag{1.1}$$

where G' is the proposal BN, $P(G|D)$ is the posterior probability of a graph given a database of cases D, and $\#nbd(G)$ is the number of neighbors which consist of the current graph G, and a set of graphs with either one edge more or one edge fewer than a current graph.

While the original version of MCMC generally performs well in little spaces with a few variables, it is rather slow in convergence with a larger number of variables, and the chain is getting trapped in local high probability. For overcoming this problem, some methods have been introduced. Friedman and Koller (2003) proposed a variety of the MCMC algorithm based on the node ordering. Niinimaki et al. (2012) presented an algorithm based on the partial node ordering to make smoother sampling space. Su and Borsuk (2016) improved the structure of MCMC for BNs through the Markov blanket resampling and Goudie and Mukherjee (2016) used a specific Gibbs sampling based on the entire sets of parents for multiple nodes from the appropriate conditional distribution.

In this paper, we focus on the discrete variables and introduce a novel Gibbs sampler for learning BN without considering entire sets of parents. More details are presented in section 2.

This paper is organized as follows: in section 2, we introduce the proposed method. The Experimental Results and Discussion are presented in section 3 and section 4, respectively.

2. Proposed Method

Suppose we have a domain of discrete variables $U = \{X_1, \cdots, X_n\}$ and a complete database of cases D. We wish to determine the joint distribution of data and BN. For this purpose, the following assumptions must be considered (Heckerman and Geiger , 1995).

Assumption 1. *Given domain U and D, let D_l denote the first $l-1$ cases in the database. In addition, let x_{il} and Π_{il} denote the variable x_i, and the parent set Π_i in the lth case, respectively. Then, for a Bayesian network structure B in U, there exist positive parameters Θ such that, for $i = 1, \cdots, n$, and for all $k, k_1, \cdots, k_{i-1}$,*

$$P(x_{il} = k | x_{1l} = k_1, \cdots, x_{(i-1)l} = k_{i-1}, D_l, \Theta) = \theta_{ijk} \tag{2.2}$$

where j is the state of Π_{il} consistent with $\{x_{1l} = k_1, \cdots, x_{(i-1)l} = k_{i-1}\}$.

This assumption is known as "Multinomial Sample Assumption" and θ_{ijk} denotes the multinomial parameters. If we let N_{ijk} be the number of cases in the database D in which $x_i = k$ and $\Pi_i = j$, then

$$P(D|\Theta) = \prod_i \prod_j \prod_k \theta_{ijk}^{N_{ijk}}. \tag{2.3}$$

Assumption 2. *Let define $\Theta_{ij} = \cup_{k=1}^{r_i} \theta_{ijk}$, $\Theta_i = \cup_{j=1}^{q_i} \Theta_{ij}$ and $\Theta = \cup_{i=1}^{n} \Theta_i$, in which r_i is the number of states of variable x_i and q_i is the number of states of Π_i. Then the proper prior distribution of Θ_{ij} is Dirichlet. This assumption says that there exist exponents N'_{ijk}, which depend on a given network B, that satisfy*

$$\pi(\Theta_{ij}|B) = c. \prod_k \theta_{ijk}^{N'_{ijk}-1}, \tag{2.4}$$

where c is a normalization constant.

When every parameter set of B has a Dirichlet distribution, we simply say that $\pi(\Theta|B)$ is also Dirichlet. Combining the Dirichlet assumption and Eq.2.2, the following posterior probability is obtained:

$$\pi(\Theta|D, B) = c. \prod_i \prod_j \prod_k \theta_{ijk}^{N'_{ijk}+N_{ijk}-1}. \tag{2.5}$$

As shown, all the above equations are calculated for a given network B. However, in practice, we imagine that the data is a random sample from an unknown B. Thus, instead of $\pi(\Theta|D, B)$, we need to determine the posterior distribution $\pi(\Theta, B|D)$. The search space of all BN structures is extremely large. It has been shown that the number of different structures grows super-exponential with respect to the number of nodes (Pearl , 1998). Thus, identifying the correct structure among all structures is a NP-hard problem. According to the product rule of probability, we have:

$$\pi(\Theta, B|D) = \pi(\Theta|B, D)\pi(B|D). \tag{2.6}$$

To sample from the joint posterior distribution Θ and B, we first sample from the posterior $\pi(B|D)$ and replace it in full conditional posterior $\pi(\Theta|B, D)$. We then have samples of Θ and B.
Our approach for estimating the marginal posterior $\pi(B|D)$ is Gibbs sampling. Gibbs sampler involves ordering the parameters and sampling from the conditional distribution for each parameter given the current value of all the other parameters and repeatedly cycling through this updating process. For learning BN using Gibbs sampler, we redefine the unknown structure B by a set of new parameters $e = \{e_{uk}, 1 \leq u < k \leq n\}$ as follows:

$$e_{uk} = \begin{cases} 0 & \text{There is no edge between node } u \text{ and node } k \\ 1 & \text{There is a dir ected edge from node } u \text{ to node } k \\ -1 & \text{There is a directed edge from node } k \text{ to node } u \end{cases} .$$

In other words, the existence and the direction of the edges in B are specified by a set of parameters $\{e_{uk}\}$. This means that

$$\pi(B|D) = \pi(\{e_{uk}\}|D). \tag{2.7}$$

The samples from $\pi(\{e_{uk}\}|D)$ represent the estimated BN. Thus, we need to calculate the following full conditional probabilities in the context of Gibbs sampling:

$$\pi(e_{uk}|\{e_{-uk}\}, D) \propto P(D|\{e_{uk}\})\pi(e_{uk}|\{e_{-uk}\}), \tag{2.8}$$

where $\{e_{-uk}\}$ is a set of edges except e_{uk}. For calculating $P(D|\{e_{uk}\})$ in Eq.2.7, we need to use the ”Parameter Independence” (Assumption 3) and ”Parameter Modularity” (Assumption 4) as follows: (Heckerman and Geiger , 1995).

Assumption 3. *Given a network structure B, we have:*

$$a.\pi(\Theta|B) = \prod_{i=1}^{n} \pi(\Theta_i|B)$$

$$b.\pi(\Theta_i|B) = \prod_{i}^{q_i} \pi(\Theta_{ij}|B).$$

Assumption 3a says that the parameters associated with each variable in a network structure are independent. This assumption is called global parameter independence. Assumption 3b says that the parameters associated with each state of the parents of a variable are independent. This assumption is called local parameter independence.

Assumption 4. *Given two network structures B_1 and B_2, if X_i has the same parents in B_1 and B_2, then:*

$$\pi(\Theta_{ij}|B_1) = \pi(\Theta_{ij}|B_2)$$

This assumption says that the densities for parameters Θ_{ij} depend only on the structure of the network.
Based on the consequences of these assumptions, the following formula is obtained for $P(D|\{e_{uk}\})$ (for more detail see (Heckerman and Geiger , 1995)):

$$P(D|\{e_{uk}\}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ijk})}{\Gamma(N'_{ijk} + N_{ij})} . \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})}. \tag{2.9}$$

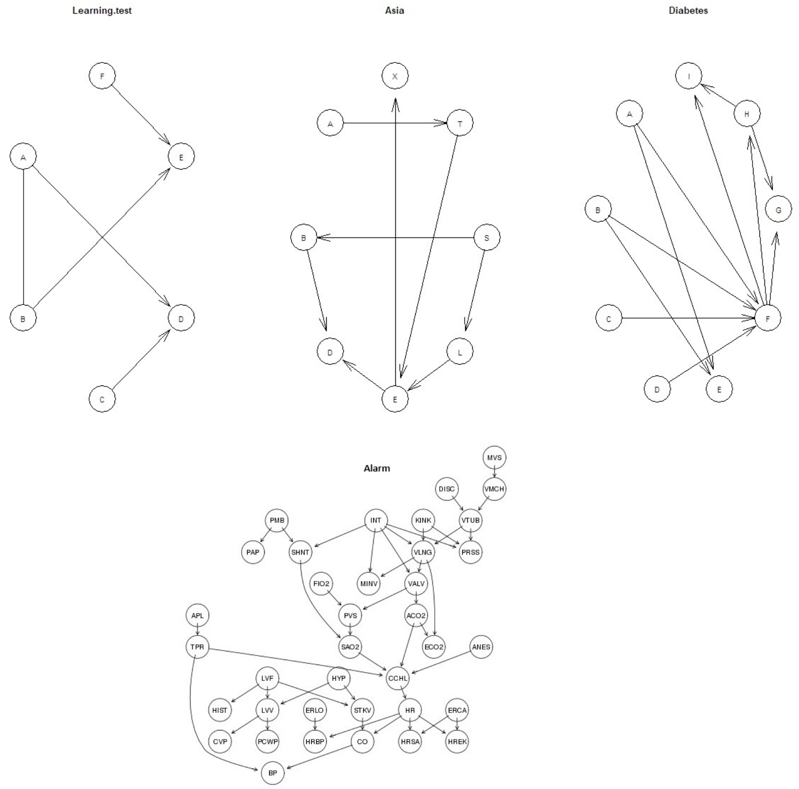
To determine the posterior distribution in Eq.2.7, we also need to determine the prior distribution $\pi(e_{uk}|\{e_{-uk}\}$ where it is a critical issue in Bayesian analysis. Most of the prior information about parameters is unreliable. This has led us to use a noninformative discrete uniform prior. More precisely, we consider the noninformative discrete uniform prior on all acyclic networks. This means that for $\pi(e_{uk} = c|\{e_{-uk}\}), c = -1, 0, 1$, if all values of c will result in an acyclic network, then:

$$\pi(e_{uk} = c|\{e_{-uk}\}) = \frac{1}{3}, \tag{2.10}$$

and if for some values of c, the graph becomes cyclic, then the prior probability $\pi(e_{uk} = c|\{e_{-uk}\})$ is uniformly distributed only on the other values of c that make the graph acyclic.
Regarding Eq.2.8 and Eq.2.9, we can take samples from the posterior $\pi(\{e_{uk}\}|D)$. These samples indicate the BNs at iterations of Gibbs sampling. Finally, for achieving a single graph that represents the best graph fitted on data, we use the mode of burn-in graphs. This means that the most frequent edges of burn-in graphs are considered as the best single graph.

3. Experimental Results

In this section, we present experimental results. We use four well-known BN structures; Asia (Lauritzen and Spiegelhalter , 1998), Diabetes (Ripley , 2007), Learning.test and Alarm (Beinlich et al. , 1989).


Learning.test

Asia

Diabetes

Alarm

Figure 1: Learning.test, Asia, Diabetes and Alarm networks

We generate 20000 cases from each BN in order to perform multiple tests and estimate more precise metrics.

- The Learning. Test is small synthetic network, for testing purposes. This BN has 6 nodes and 5 edges; each node has two or three values. This BN is available in the "bnlearn" package in R.
- The Asia network has 8 nodes and 8 edges; each variable one has two attributes. This BN is a small BN about lung diseases (tuberculosis, lung cancer, or bronchitis) and visits to Asia.
- The Diabetes network has 9 nodes and 11 edges. This BN is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict whether a patient has diabetes based on diagnostic measurements.
- The ALARM network has 37 variables and 46 edges; each variable has two, three, or four possible attributes. The Alarm network is designed to provide an alarm message system for patient monitoring.

The existence of the original BNs allows us to define important terms, which indicate the performance of our approach. In this line, we compare the edge scores by computing the number of correct edges, missing, reverse, and additional compared to the original BN by the following definitions:

- Correct edge (C): Edges detected with the same edge direction.
- Reverse edge (R): Edges detected with the opposite edge direction.
- Missing edge(M): Edges not detected compared to the original BN.
- Additional edge(A): Edges that are not presented in the original BN.

Based on these definitions, graph error edges (E) are computed by summing the three erroneous edge types (i.e., missing, reverse, and additional edges). The edge scores make it possible to define the performance accuracy of the proposed approach by the following equation:

$$Accuracy = \frac{C}{C+E}. \tag{3.11}$$

As it is known, in the process of the Gibbs sampling, we obtain many BNs. Thus, for achieving a single graph that represents the best graph fitted on data, we use the mode of burn-in graphs. To evaluate the final graph obtained by the proposed method, we report the True Positive rate (TPR) and False Positive Rate, which are defined as follows (Fu , 2012):

$$TPR = \frac{C}{T}, \qquad FPR = \frac{R + A}{n(n-1) - T} \tag{3.12}$$

where T is the total number of original edges. A higher value of $TPRs$ and small values of $FPRs$ represent the proper graphs fitted to the data.
We compare the performance of our approach with the original MCMC, the method proposed by Goudie and Mukherjee (2016), and the following well-known approaches:

- Max-Min Hill-Climbing (MMHC): The MMHC is a hybrid algorithm that combines the Max-Min parents and children algorithm to restrict the search space and the Hill-Climbing algorithm to find the optimal graph structure in the restricted space (Tsamardinos et al. , 2006). The algorithm first identifies the parents and children set of each variable, then performs a greedy hill-climbing search in the space of DAGs. The search begins with an empty graph. The edge addition, deletion, or direction reversal that leads to the largest increase in score is taken, and the search continues similarly recursively.

- 2-phase Restricted Maximization (RSMAX2): The RSMAX2 is a more general implementation of the Max-Min Hill Climbing, which can use any combination of constraint-based and score-based algorithms (Tsamardinos et al. , 2006).

The results of using these methods are shown in 1. Table1 shows that the proposed method has higher correct values and fewer errors. The results of this paper are very close to the results of Goudie and Mukherjee (2016). The computational complexity of the proposed method is higher than the Goudie and Mukherjee (2016). For instance, when the number of nodes is 9 (Diabetes), the estimated graphical model for the proposed method contains 11 directed edges, of which 9 edges are present in the true graph, one edge has direction reversed, and one edge is not included in the true graph.

Table 1: Comparing Edge Scores, TPR and FPR

Data	Edge Type	Proposed Method	Original MCMC	MMHC	RSMAX2	Goudie et al.
Learning.test	Correct	4	3	4	4	4
	Reverse	1	1	1	1	1
	Additional	0	1	1	0	0
	Missing	0	1	0	1	0
	Graph error	1	3	2	2	1
	TPR	0.80	0.60	0.80	0.80	0.80
	FPR	0.04	0.08	0.08	0.08	0.04
Asia	Correct	8	5	6	4	8
	Reverse	0	1	0	0	0
	Additional	0	0	1	0	0
	Missing	0	1	2	4	0
	Graph error	0	2	3	4	0
	TPR	1	0.62	0.75	0.5	1
	FPR	0	0.02	0.02	0	0
Diabetes	Correct	9	6	9	7	9
	Reverse	1	1	1	1	1
	Additional	0	1	1	3	1
	Missing	1	2	1	3	1
	Graph error	2	4	3	7	3
	TPR	0.81	0.54	0.81	0.63	0.81
	FPR	0.01	0.02	0.02	0.04	0.02
Alarm	Correct	39	26	32	32	40
	Reverse	1	6	6	1	1
	Additional	0	8	5	0	0
	Missing	2	7	8	13	2
	Graph error	3	21	19	14	2
	TPR	0.84	0.56	0.69	0.69	0.86
	FPR	0.0007	0.01	0.008	0.0007	0.0007

4. Conclusion

As it is known, the number of BN structures is super-exponential in the number of random variables in the domain. Consequently, the summation of all possible structures can be computed in a closed form only for small domains, or those with supplemental constraints that restrict the space. In this paper, we focus on the small BNs with a few discrete variables and use a Gibbs sampler for learning BN. The results in Figure 3 suggest that our method can estimate the structures of BNs with reasonable accuracy. For further work, we use the data information to restrict the space of possible graphs. This means that we make zero prior probability of some possible graphs and then apply the Gibbs sampler. This perspective makes it possible to reduce the search space in the process of the MCMC simulation.

References

Arias, J., Gámez, J. A. and Puerta, J. M. (2015). Structural learning of bayesian networks via constrained hill climbing algorithms: Adjusting trade-off between efficiency and accuracy. *International Journal of Intelligent Systems*, **30(3)**, 292–325.

Beinlich, I.A., Suermondt, H.J., Chavez, R.M. and Cooper, G.F. (1989). *The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks.* Springer.

Friedman N., and Koller, D. (2003). Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, **50(1-2)**, 95–125.

Fu, F. (2012).*Sparse causal network estimation with experimental intervention.* University of California, Los Angeles, USA.

Giudici, P., and Castelo, R. (2003). Improving markov chain monte carlo model search for data mining. *Machine learning*, **50(1-2)**, 127–158.

Goudie, R.J., and Mukherjee, S. (2016). A gibbs sampler for learning dags. *The Journal of Machine Learning Research*, **17(1)**, 1032–1070.

Heckerman, D. (1998) A tutorial on learning with bayesian networks. *Learning in graphical models*, Springer, 301–354.

Heckerman, D. and Geiger, D. (1995). Learning bayesian networks: a unification for discrete and gaussian domains. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 274–284 1995.

Lauritzen, S.L., and Spiegelhalter, D.J. (1998). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, **50(2)**, 157–224.

Madigan, D., York, J., and Allard, D. (1995). Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, **63(2)**, 215–232.

Niinimaki, T., Parviainen, P., and Koivisto, M. (2012), Partial order mcmc for structure discovery in bayesian networks. *arXiv preprint arXiv:1202.3753.*

Pearl, J. (1998). *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufman.

Ripley, B.D. (2007). *Pattern recognition and neural networks.* Cambridge university press.

Su, C., and Borsuk, M.E. (2016). Improving structure mcmc for bayesian networks through markov blanket resampling. *The Journal of Machine Learning Research*, **17(1)**, 4042–4061.

Tsamardinos, I., Brown, L.E., and Aliferis, C.F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, **65(1)**, 31–78.