Online Learning of Bayesian Network Parameters with Incomplete Data

Sungsoo Lim and Sung-Bae Cho

Dept. of Computer Science, Yonsei University
Shinchon-dong, Seodaemun-ku,
Seoul 120-749, Korea
lss@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. Learning Bayesian network is a problem to obtain a network that is the most appropriate to training dataset based on the evaluation measures given. It is studied to decrease time and effort for designing Bayesian networks. In this paper, we propose a novel online learning method of Bayesian network parameters. It provides high flexibility through learning from incomplete data and provides high adaptability on environments through online learning. We have confirmed the performance of the proposed method through the comparison with Voting EM algorithm, which is an online parameter learning method proposed by Cohen, *et al.*

1 Introduction

The parameters of a Bayesian network (BN) are determined by the use of expert opinion or by learning from data [1]. The former has the advantage of reflecting experts' knowledge, but it is a difficult and time-consuming process. Moreover, it is not clear whether the network designed by the experts is really the most appropriate model for the domain. Although the latter, learning from data, can overcome the problems of the former, it is not always available because the data cannot be ready at all the time the BN is constructed. Furthermore, it cannot consider the change of environments.

To overcome these limitations, online learning methods are discussed [2], [3]. Online learning of BN parameters is a method that learns parameters of BN using the given data and parameters at time *t*. Cohen, *et al.* proposed Voting EM algorithm which adopts EM(η) algorithm to online learning [4], [5]. Zhang, *et al.* verified the usefulness of Voting EM algorithm by utilizing it to flood decision-supporting system [6].

Although, in the case of complete data, Voting EM algorithm can quickly converge to the proper parameters, in the case of incomplete data, it adapts parameters partially and incorrectly. It is because Voting EM algorithm is based on EM algorithm. EM algorithm can estimate the missing data by using many data, but Voting EM algorithm, which uses only one data, cannot do well. In this paper, we propose a novel method for online BN parameter learning which can overcome the limitation of Voting EM algorithm.

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNAI 4114, pp. 309 – 314, 2006.

2 Voting EM Algorithm

Let x_i be a node in the network that takes any value from the set $\{x_i^1, x_i^2, \cdots, x_i^p\}$ and π_i be the set of parents of in the network that takes one of the configurations denoted by $\{\pi_i^1, \pi_i^2, \ldots, \pi_i^p\}$ then we can define an entry in the CPT (Conditional Probability Table) of the variable x_i as $\theta_{ijk} = P(x_i = x_i^k \mid \pi_i = \pi_i^j)$. Online learning of BN parameters is to get the new set of parameters Θ_{t+1} from the given set of parameters Θ_t and the observed data d_t at time t as follows:

$$\Theta_{t+1} = \arg\max_{\Theta}[\eta L(\Theta \mid D) - d(\Theta, \Theta_t)] \cdot \tag{1}$$

Where $L(\Theta|D)$ denotes log likelihood and $d(\Theta, \Theta_t)$ denotes the distance between the sets of parameters, Θ and Θ_t. Therefore, Θ_{t+1} is the new set of parameters which has high log likelihood with given data set D and the character of the set of parameter Θ_t. η, which is the importance of log likelihood comparing with distance factor, denotes learning rate. Bauer, *et al.* solve the maximization problem of Eq. (1) with constrains that $\Sigma_k \theta_{ijk} = 1$ for all i and j, called EM(η) algorithm [3]. Cohen, *et al.* proposed Voting EM algorithm which adapted EM(η) algorithm to online learning [4], [5].

$$\theta_{ijk}^{t+1} = \begin{cases} (1-\eta)\theta_{ijk}^t + \eta, & \text{if } x_i = x_i^k \text{ at } d_t \text{ and } P(\pi_i^j \mid d_t, \theta_t) \neq 0 \\ (1-\eta)\theta_{ijk}^t, & \text{if } x_i \neq x_i^k \text{ at } d_t \text{ and } P(\pi_i^j \mid d_t, \theta_t) \neq 0 \\ \theta_{ijk}^t, & \text{otherwise} \end{cases} . \tag{2}$$

As shown in the Eq. (2), Voting EM algorithm only learns the parameter θ_{ijk}^{t+1} when the node x_i is observed. Moreover, even though the node x_i is observed, if all of the parent nodes of x_i are not observed, it does not learn correctly because it does not consider the relation between the node x_i and unobserved parent nodes.

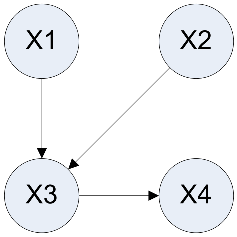


Fig. 1. Simple Bayesian network

For example, if there exists Bayesian network like Fig. 1 and if the node x_3 is unobserved node ($d_t = \{x_1^a, x_2^b, x_3^?, x_4^d\}$), the parameters at the node x_3 are not learned and the parameters at the node x_4 are learned without considering the effect of the node x_3 as follows: In the case of $P(x_3^j \mid d_t, \Theta_t) \neq 0$, if $x_4^k \neq x_4^d$ then $\theta_{4jk}^{t+1} = (1-\eta)\ \theta_{4jk}^t$ and if $x_4^k = x_4^d$ then $\theta_{4jk}^{t+1} = (1-\eta)\ \theta_{4jk}^t + \eta$. Otherwise, $\theta_{4jk}^{t+1} = \theta_{4jk}^t$. In other words, if the node x_3 is unobserved, the node x_4 only learns the probability $P(x_4 = x_4^d \mid d_t, \Theta_t)$ and it cannot correctly learn the relation between x_3 and x_4 which means $P(x_4 = x_4^d \mid \pi_4^j, d_t, \Theta_t)$. In this paper, we propose a novel method to overcome such limitations of Voting EM algorithm which can learn BN parameters with incomplete data.

3 Proposed Learning Method

The data used for learning BN parameters online are the observed value $\hat{x}_i^k$, which denotes the value of observed state of each node x_i, and the predicted value $P(x_i = x_i^k \mid d_t - \{x_i\}, \Theta_t)$, which shows how well the set of parameters Θ_t can predict the situation d_t. The value of $\hat{x}_i^k$ is 1 when the state of node x_i is k; otherwise, it becomes 0. Using these data, we first find the set of parameters $\overline{\Theta}$ well fit for the data d_t, and get the next set of parameters Θ_{t+1} by exponential smoothing method as following equation:

$$\forall\, ijk, \theta_{ijk}^{t+1} = (1-\eta)\theta_{ijk}^{t} + \eta\overline{\theta}_{ijk}\,. \tag{3}$$

where η denotes the rate of convergence.

Now let us find out how we get the set of parameters $\overline{\Theta}$. If we have the Bayesian network structure that consists of n nodes, let $O = \{x_{O_1}, x_{O_2}, \cdots, x_{O_a}\}$ be the set of observed nodes except x_i and $U = \{x_{U_1}, x_{U_2}, \cdots, x_{U_{n-a-1}}\}$ be the set of unobserved nodes except x_i. Then we can get the predicted value $P(x_i = x_i^k \mid d_t - \{x_i\}, \Theta_t)$ by Eq. (4) and it can be rewritten as Eq. (5), which consists of CPT variables, by using the independent assumption and chain rules.

$$P(x_i = x_i^k \mid d_t - \{x_i\}, \Theta_t) = \frac{\sum\limits_{\text{for all state of } x\in U} P(x_{o_1} = x_{o_1}^{s_1}, \cdots, x_{o_a} = x_{o_a}^{s_a}, x_i = x_i^k, x_{u_1}, \cdots, x_{u_{n-a-1}})}{\sum\limits_{\text{for all state of } x\in U\cup\{x_i\}} P(x_{o_1} = x_{o_1}^{s_1}, \cdots, x_{o_a} = x_{o_a}^{s_a}, x_i, x_{u_1} \cdots, x_{u_{n-a-1}})}\,. \tag{4}$$

$$\frac{\sum\limits_{\text{for all state of } x\in U} P(x_{o_1} = s_1 \mid \pi_{o_1}) \times \cdots \times P(x_{o_a} = s_a \mid \pi_{o_a}) \times P(x_i \mid \pi_i) \times P(x_{u_1} \mid \pi_{u_1}) \times \cdots \times P(x_{u_{n-a-1}} \mid \pi_{u_{n-a-1}})}{\sum\limits_{\text{for all state of } x\in U\cup\{x_i\}} P(x_{o_1} = s_1 \mid \pi_{o_1}) \times \cdots \times P(x_{o_a} = s_a \mid \pi_{o_a}) \times P(x_i \mid \pi_i) \times P(x_{u_1} \mid \pi_{u_1}) \times \cdots \times P(x_{u_{n-a-1}} \mid \pi_{u_{n-a-1}})}\,. \tag{5}$$

Through canceling, we can simplify Eq. (5) to A / (A + B), where A and B consist of the sum of multiplication of CPT variables. Therefore, we can update the parameters as the following. If the observed value $\hat{x}_i^k$ is 1, it increases A and decreases B; if the value is 0, it decreases A and increases B. However, there are too many factors consisting of A and B, and so it requires much computational time to apply for all i and k. We assume that only the parent nodes of x_i affect the node x_i and the affected parent nodes are independent. With these assumptions, we can rewrite Eq. (5) as Eq. (6).

We update the parameters, which are related with the predicted value $P(x_i = x_i^k \mid d_t - \{x_i\}, \Theta_t)$ by using Eq. (6) and the value of observed node $\hat{x}_i^k$ according to the weight to the predicted value.

$$\begin{aligned} P(x_i = x_i^k \mid d_t - \{x_i\}, \Theta_t) &= \sum_{\text{for all } j} P(x_i = x_i^k \mid \pi_i^j, d_t - \{x_i\}, \Theta_t) P(\pi_i^j \mid d_t - \{x_i\}, \Theta_t) \\ &= \sum_{j\in P(\pi_i^j \mid d_t - \{x_i\}, \theta^t) \neq 0} \theta_{ijk}^t \prod_{x_a^b \in \pi_i^j} P(x_a^b \mid d_t - \{x_i\}, \Theta_t) \end{aligned} \tag{6}$$

The following pseudo code shows the whole process of the proposed method.

Pseudo code of proposed method

```
procedure OnlineLearning($d_t$)
begin
  for all $x_i \in S$   //$S$ is the structure of BN
    for $k$:=0 to $p_i$
      if $\hat{x}_i^k$:=1 then Update($i$,$k$,1.0);
      else if $\hat{x}_i^k$:=0 then Update($i$,$k$,-1.0);
  for all $\omega_{ijk} \in \Omega$  //$\Omega$ is a set of weights for updating ${}_t$
    if $\omega_{ijk} > 0$ then $\bar{\theta}_{ijk}$ := (1- $\omega_{ijk}$) + ${}_{ijk}^{t}$+ $\omega_{ijk}$;
    else $\bar{\theta}_{ijk}$ := (1- $\omega_{ijk}$) + ${}_{ijk}^{t}$;
  for all $i$, $j \in \bar{\Theta}$
    normalize to $\Sigma_{\forall k}\bar{\theta}_{ijk}$=1;
  for all $\bar{\theta}_{ijk} \in \bar{\Theta}$
    ${}_{ijk}^{t+1}$:= (1-$\eta$) ${}_{ijk}^{t}$ + $\eta\bar{\theta}_{ijk}$;
end

procedure Update($i$,$j$,$w$)
begin
  if $w < threshold$ then return;
  for all $\pi_i^j \in S$
    $\omega_{ijk} := \omega_{ijk} + wP({}_i^j | d_t - \{x_i\}, {}_t)$;
  for all $\pi_i^j \in S$
    if $P({}_i^j | d_t - \{x_i\}, {}_t) \neq 0$ then
      for all $x_a^b \in {}_i^j$ and $x_a$ at $d_t$=null
        Update($a$, $b$, $w$ ${}_{ijk}^{t} \prod_{x_c^d \in \pi_i^j - \{x_a^b\}} P(x_c^d | d_t - \{x_i\}, {}_t)$);
end
```

4 Experiments

We have conducted a comparison test with Voting EM algorithm at Asia network to manifest the performance of the proposed method. We have collected 10,000 data as the samples of learning data: To test the adaptability, we get the first 5,000 of the data from the real network and the last 5,000 of the data from the modified network where the probability of the attack of tubercle when he or she visits Asia to 40%.

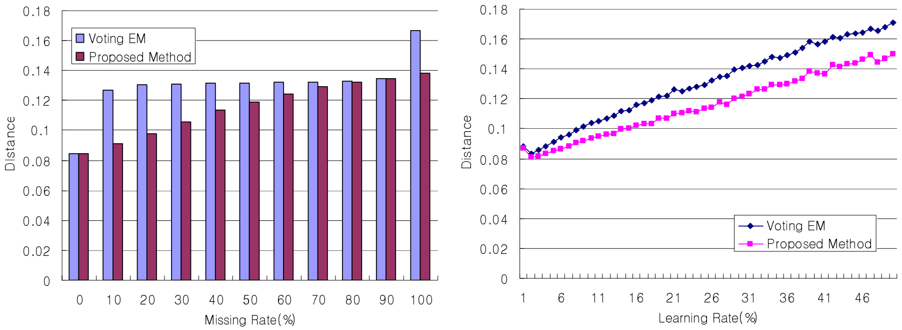
The comparison test has been conducted by changing the missing rate of data and the learning rate. The missing rate that is unobserved value at the tuberculosis node is changed from 0% to 100% with 10% intervals and the learning rate is changed from 1% to 50% with 1% intervals. We evaluate the learned parameters using the distance between the learned parameter and real parameter as follows:

$$D(\theta,\theta') = \sum_{\text{for all } i,j,k} |\,\theta_{ijk} - \theta'_{ijk}\,| \cdot \tag{7}$$

Fig. 2 shows the distance of Voting EM algorithm and the proposed method from the real parameters in terms of missing rate and learning rate. As shown in Fig. 3(a), when the given data is complete, both algorithms manifest the same performance:

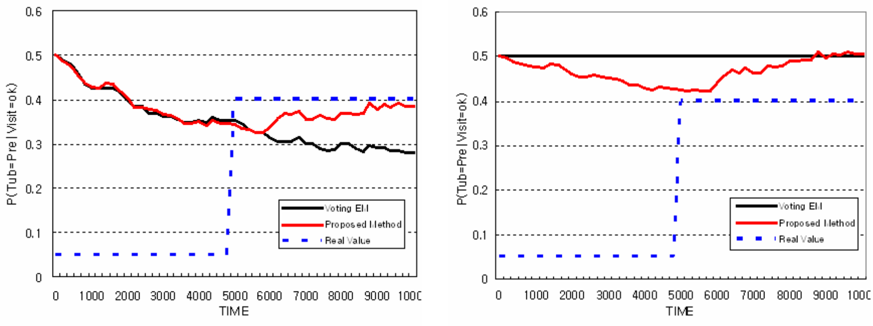
When the data is complete, the proposed method is the same as the Voting EM algorithm. However, when the data is incomplete, Voting EM algorithm shows great difference when compared with the result of the complete data. It shows that Voting EM algorithm is weak when the incomplete data is used. Especially, when the missing rate is 100% (no data at tuberculosis node), it has learned nothing. On the other hand, the proposed method is robust at incomplete data. In the case of learning rate, Fig. 2(b) shows that the proposed method performs better than Voting EM algorithm. Moreover, it shows that the learning 2% provides the best performance.

Fig. 3 shows the convergence of the proposed method and Voting EM algorithm to real value according to the missing rate at tuberculosis node. As shown in Fig. 3(a), the proposed method provides better performance than Voting EM algorithm and Fig. 3(b) shows that the proposed method can acknowledge the change of environments though Voting EM algorithm learns nothing when the missing rate is 100%. Although it seems that the proposed method shows worse performance than Voting EM algorithm after time 9000 in Fig. 3(b), if we compare it with all the other parameters, the proposed method performs better than Voting EM algorithm as shown Fig. 2(a).



(a) Result through the missing rate (b) Result through the learning rate

Fig. 2. Result of comparison test



(a) Missing Rate 50% (b) Missing Rate 100%

Fig. 3. Convergence to real value at node *tuberculosis* using $\eta = 0.02$

5 Concluding Remarks

Online BN learning is needed for modeling the variable environment or the change of user's preference. Voting EM algorithm is one of good methods to learn BN parameters online. However, there are limits to its capability in learning with the incomplete data. In this paper, we have proposed an online learning method of BN parameters which works well with incomplete data. We have confirmed the performance of the proposed method through the comparison with Voting EM algorithm.

For the future works, it is required to analyze the convergence of the proposed method and conduct more experiments in realistic domains. In addition, by adding a mechanism in order to automatically adjust the learning rate η, we can obtain an enhanced version of the online learning method that learns more quickly than the present method.

Acknowledgments

This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Commerce, Industry and Energy.

References

1. Heckerman, D.: A Tutorial on Learning with Bayesian Networks. In Report No. MSR-TR-95-06, Microsoft Research (1995)
2. Spiegelhalter, D., Lauritzen, S.: Sequential Updating of Conditional Probabilities on Directed Graphical Structures. Networks, 20 (1990) 579-605
3. Bauer, E., Koller, D., Singer, Y.: Update Rules for Parameter Estimation in Bayesian Networks. Proceedings of the 13th. Annual Conference on Uncertainty in AI (1997) 3-13
4. Cohen, I., Bronstein, A., Cozman, F. G.: Online Learning of Bayesian Network Parameters. In Report No. HPL-2001-55, HP Labs (2001)
5. Cohen, I., Bronstein, A., Cozman, F. G.: Adaptive Online Learning of Bayesian Network Parameters. In Report No. HPL-2001-156, HP Labs (2001)
6. Zhang, S. Z., Yu, H., Ding, H., Yang, N. H., Wang, X. K.: An Application of Online Learning Algorithm for Bayesian Network Parameter. Machine Learning and Cybernetics, 1 (2003) 153-156