Incremental Learning of Bayesian Networks from Concept-Drift Data

Haibo Yu
China Electric Power Research Institute
Beijing, China
e-mail: yuhaibo_cepri@126.com

***Abstract*—Bayesian network utilizes graphical model to describe dependencies among variables in probabilistic way, it is one of the most important model for uncertainty processing in Artificial Intelligence. Incremental learning of Bayesian networks has been received more attentions in recent years, in this paper a novel method is proposed to learn Bayesian network from incremental data. In this method, a novel incremental scoring function is designed to adaptively adjust the tendency of matching new and old data in the process of incremental learning. We propose an improved adaptive incremental structure learning algorithm for Bayesian network. Theoretical analysis and experimental results both demonstrate the proposed method outperforms other state-of-the-art methods.**

Keywords-Bayesian network; incremental learning; machine learning; parameter learning

I. INTRODUCTION

With the coming of big data era, the statistical machine learning for probabilistic graphical model has attracted extensive attention in recent years. Bayesian network is one of the most typical probabilistic graphical models which is a fundamental model for uncertainty processing in Artificial Intelligence [1].

Learning Bayesian networks from data is NP-hard problem and still one of the most challenges in machine learning [2]. The incremental learning of Bayesian networks is an area that has gained more importance in recent years, in this case, data records are received sequentially, and Bayesian network is constructed incrementally [3]. In this paper, we propose a score-based adaptive algorithm to learn Bayesian network in the presence of concept drift. We design a scoring function which makes the learning process adaptively regulate the searching strategy for each local structure of Bayesian network, then we propose an adaptive parameter learning method based on Lagrange multiplier, we also provide an improved structure learning method.

The remainder of this paper is organized as follows: Section II provides the preliminaries and notations of the incremental learning of Bayesian network, and the novel scoring function is proposed in Section III. Section IV offers learning method. Section V offers the experimental results and comparisons of the proposed method. Finally, conclusions are summarized in Section VI.

II. PRELIMINARIES AND NOTATION

For a set of variables $X=\{X_1,X_2,\ldots,X_n\}$, a Bayesian network is composed of graph structure G and parameters Θ; in which G is a directed acyclic graph (DAG), and each node of the graph represents the variable, edges represent direct dependencies between variables; the parent node set of variable X_i is denoted by π_i, then $\Theta=\bigcup_{i=1}^{n}\{P(X_i\mid\pi_i)\}$ represents the conditional probability distribution of each node given the values of their parent nodes in the network. Assumed the range of X_i is $\{x_i^1,\ldots,x_i^{r_i}\}$, the range of π_i is $\{\pi_i^1,\ldots,\pi_i^{q_i}\}$, the local conditional probability distribution of each node is represented by $\theta_{ijk}=P(X_i=x_i^k\mid\pi_i=\pi_i^j)$, and $\Theta=\bigcup_{i=1}^{n}\bigcup_{j=1}^{q_i}\bigcup_{k=1}^{r_i}\{\theta_{ijk}\}$.

The problem of incremental learning of Bayesian networks can be stated as follows: the pre-existing data (old data) is presented as $D'=\{C_1,C_2,\ldots,C_{N'}\}$, N' is the size of D', that means the old data contains N'samples (observations), and the current Bayesian network which is learned from D' is represented as B'. The incoming new data is presented as $D=\{C_{N'+1},C_{N'+2},\ldots,C_{N'+N}\}$, N is the size of D, i.e. the new data contains N samples. The aim of incremental learning is to learn new Bayesian network B using B', $D'$and D, namely, update B' to B with D.

III. DESIGN OF SCORING FUNCTION

The requirement of incremental learning is that the learned Bayesian network should fit (reflect or match) both old data and new data. That is to say the scoring function reflects not only the fitness between current Bayesian network and old data, but also the fitness between current Bayesian network and new data. In addition, the more complex Bayesian network is more difficult to utilize because of the high inference complexity, so the learning algorithm should tend to learn concise network structure. For the above requirements, log-likelihood $\log P(D'\mid B)$ is used to measure the fitness between a Bayesian network B and old data D', and log-likelihood $\log P(D\mid B)$ is used to measure the fitness between a Bayesian network B and old data D.

If the scoring function only uses $\log P(D'\mid B)$ and $\log P(D\mid B)$, then during the process of incremental learning, the proportion of $\log P(D'\mid B)$ will grow large gradually with the collection of old data. So in order to ensure the fairness of old and new data in the learning process, we adopt $\log P(D'\mid B)/N'$ and $\log P(D\mid B)/N$ in scoring function, that is the fitness divided by the size of data

set, in other words, the fitness between Bayesian network and each sample of the data set.

Furthermore, we hope the learning algorithm is adaptive and flexible. When the new data arrives, if the current Bayesian network fits the new data well, it indicates the current network is fairly accurate, so at this moment the learning algorithm can trend to old data because the learning result need not change too much compared with the current Bayesian network. On the contrary, if the current Bayesian network fits the new incoming data poorly, it indicates the current network is not precise, thus the proportion of new data in scoring function should be enhance to make the learning algorithm update the existing Bayesian network substantially, so that the learning result can adapt the new data.

Definition 1. (Local Structure) In Bayesian network, the local structure of node X_i is the sub-structure composed by node X_i and its parents nodes π_i in the Bayesian network.

In the incremental learning algorithm, for each local structure of current existing Bayesian network B', we introduce an adaptive tendency factor η to adjust the learning tendency to old or new data. Based on above discussion, we proposed the novel scoring function for adaptive incremental learning of Bayesian network with Definition 2.

Definition 2. (Scoring Function). For new data D and old data D', the scoring function of a Bayesian network B is:

$$\begin{aligned} &Score(B:D,D') \\ &= \eta \cdot \frac{1}{N'} \log P(D' \mid B) + (1-\eta) \cdot \frac{1}{N} \log P(D \mid B) - Pen(B) \\ &= \frac{1}{N'} \sum_i \sum_j \sum_k \eta_i \cdot N'_{ijk} \log \theta_{ijk} + \frac{1}{N} \sum_i \sum_j \sum_k (1-\eta_i) \cdot N_{ijk} \log \theta_{ijk} \\ &- \frac{\log(N'+N)}{2} \sum_i \| \pi_i \| (\| X_i \| - 1) \end{aligned} \quad (1)$$

where N_{ijk} is the number of cases in D in which $X_i = x_i^k$ and $\pi_i = \pi_i^j$, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, and θ is parameter.

It can be seen from Definition 2 that if η_i >0.5, the fitness between current Bayesian network and old data has larger proportion in scoring function, so the learning process trends to old data. If η_i <0.5, the fitness between current Bayesian network and new data has larger proportion in scoring function, so the learning process trends to new data. The $Pen(B)$ is a penalty function on Bayesian network B, the more complex the structure of B is, the larger the $Pen(B)$ is.

IV. Structure Learning

One of the most commonly used structure learning method for Bayesian networks is hill-climbing search. The basic idea of a conventional hill-climbing based structure learning is: First, we choose an initial network structure B_0, second, evaluate all the possible changes that can be made to B_0, and then make the change to B_0 which maximizes the scoring function, iterating above steps until there is no change can increase the scoring function. A change to a Bayesian network structure includes: add an arc (i.e. add a parent node to a node), delete an arc (i.e. delete a parent node from a node), and reverse an arc. All changes are subject to the constraint that the resulting network contains no directed cycles. Because reverse an arc $X_j \rightarrow X_i$ can be regarded as delete an arc $X_j \rightarrow X_i$ and add an arc $X_j \leftarrow X_i$, so the core changes are add an arc and delete an arc.

An efficient version of hill-climbing method is proposed by Gamez [4]. They introduced a set of forbidden parents FP(X_i) to each node X_i, the nodes in set FP(X_i) are independent of X_i, that means the nodes in FP(X_i) can not to be the parent nodes of X_i, so during the search, the nodes in FP(X_i) need not to be considered while evaluating the change that adding a possible potential parent node to X_i, thus the efficiency is improved. Their tabu list FP(X_i) is only used for adding a parent node.

In this paper, we extend the idea of Gamez and introduced two other tabu lists during the search: one is used for deleting parent node, named FDP(X_i) (abbreviated from Forbidden_Delete_Parent), the nodes in FDP(X_i) have strong dependence on X_i, so during the search, the nodes in FDP(X_i) need not to be considered while evaluating the change that deleting a parent node to X_i. Another proposed tabu list is FCY(X_i) (abbreviated from Forbidden_Cycle) which is used to avoid cycles, there will generate a cycle in network structure if add a node from FCY(X_i) to be parent of X_i. So during the search, the nodes in FCY(X_i) also need not to be considered while evaluating adding a possible parent node to X_i. For consistency, FP(X_i) is represented as FAP(X_i) (abbreviated from Forbidden_Add_Parent).

In detail, during the iteration of structure search, if adding a parent node X_j to X_i(i.e. add an arc $X_j \rightarrow X_i$) would reduce the scoring function(ΔScore<0), then in future the scoring function will also not increase by adding $X_j \rightarrow X_i$ until the local structure of X_i are changed. So X_j can be put in FAP(X_i). Similarly, if deleting parent node X_j of X_i(i.e. delete an arc $X_j \rightarrow X_i$) would increase the scoring function (ΔScore>0), that means add this arc will reduce scoring function, so put X_j into FAP(X_i). Conversely, if delete arc $X_j \rightarrow X_i$ would reduce the scoring function, that means in the subsequent iterations of search, scoring function could not increase by deleting $X_j \rightarrow X_i$ as long as the local structure of X_i are not changed, so X_j will not be considered when evaluating to delete a parent node, in this case we put X_j into FDP(X_i).

After make the change which maximizes the scoring function of current Bayesian network, the tabu lists FAP and FDP of node corresponding to the final change will be cleared and reset, in detail, if the final change is delete arc $X_j \rightarrow X_i$, that means add this arc will reduce scoring function in future, so put X_j into FAP(X_i). If the final change is add arc $X_j \rightarrow X_i$, that means afterwards delete this arc will reduce scoring function, so put X_j into FDP(X_i).

To implement cycle tabu list FCY(X), we know in fact the FCY(X) is set of descendant nodes of X, because add an arc from a descendant of X to X will generate a cycle in the Bayesian network structure. At beginning, we can obtain each FCY(X) by using *Depth-First Search* (DFS) to the graph of the Bayesian network structure with starting node X,

and the nodes in search route are the descendant of X_i to X. During the structure search, if a change adding or deleting arc $X_j \rightarrow X_i$ is made, only FCY of ancestors of X_j need update, thus we can implement that by using DFS to node X which $X_j \in$ FCY(X).

Finally, The proposed algorithm is described as Algorithm 1.

Algorithm 1 Adaptive Learning of Bayesian Networks

Input: current Bayesian network B'; new data D; old data D';
Output: new Bayesian network B;

/* Initialization*/
finish ← **false**;
$B \leftarrow B'$;
Calculate η_i for each i
for each node X in B **do**
  FAP(X_i) ←∅; FDP(X_i) ←∅;
  Initialize FCY(X) using *Depth-First Search* with starting node X;
end for
/* Structure Search*/
while finish = **false do**
 /* add an arc */
 for each node X_i and $X_j \notin (\pi_i \cup FAP(X_i) \cup FCY(X_i))$ **do**
  $\Delta Score = Score(B + \{X_j \rightarrow X_i\}) - Score(B)$;
  if $\Delta Score$ <0 **then** FAP(X_i) ← FAP(X_i)∪{ X_j };
  Store the modification which maximizes $\Delta Score$;
 end for
 /* delete an arc */
 for each node X_i and each node $X_j \in \pi_i - FDP(X_i)$ **do**
  $\Delta Score = Score(B - \{X_j \rightarrow X_i\}) - Score(B)$;
  if $\Delta Score$ <0 **then** FDP(X_i) ← FDP(X_i)∪{ X_j };
  if $\Delta Score$ >0 **then** FAP(X_i) ← FAP(X_i)∪{ X_j };
  Store the modification which maximizes $\Delta Score$;
  end for
 /* make the change */
 $m \leftarrow \underset{m \in Modifications(B)}{\arg\max} \Delta Score(B+m)$;
 $\Delta Score \leftarrow Score(B+m^*) - Score(B)$;
 if $\Delta Score > 0$ **then**
  $B \leftarrow$apply m over B;
  if m= "add $X_j \rightarrow X_i$" **then**
   FAP(X_i) ←∅; FDP(X_i) ←{ X_j };
  else if m= "delete $X_j \rightarrow X_i$" **then**
   FAP(X_i) ←{ X_j }; FDP(X_i) ←∅;
  end if
  for each X **do**
   if $X_j \in$ FCY(X) **then**
    reset FCY(X) using *Depth-First Search*;
   end if
  end for
 else finish ←**true**;
 end if
end while
return B.

V. Experimental Results

We use the dataset generated by the Alarm network [5], and compare the proposed algorithm with the algorithm of Friedman [6] and the algorithm of Alcobe [7]. The accuracy of learning results is measured by Lormalized Log-Loss. The expression is $\sum_n \frac{P^*(X)}{P_B(X)}$ [6], where $P^*(X)$ is the target probability distribution. This measurement can measure how close the learning result is to the network from which the test data is generated. The incremental algorithm reads k samples each time. We make experiments with k=200 and k=400, respectively record the results of the first 10 learning. Fig. 2 to Fig. 3 shows the experimental results. It can be observed that the proposed algorithm outperforms other algorithms in most cases.

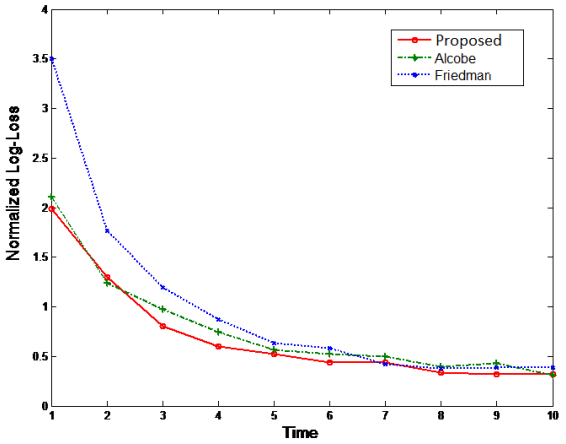

Figure 1. Comparisons of the algorithms on k=200

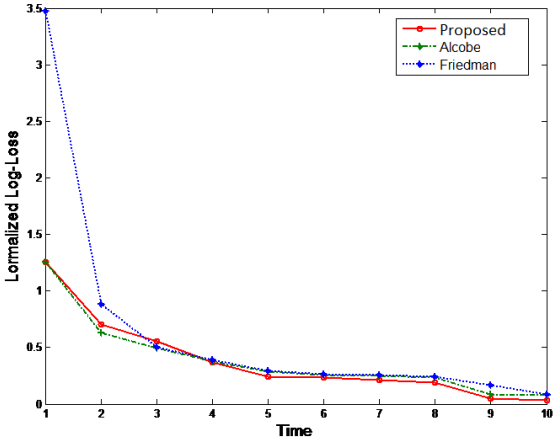

Figure 2. Comparisons of the algorithms on k=400

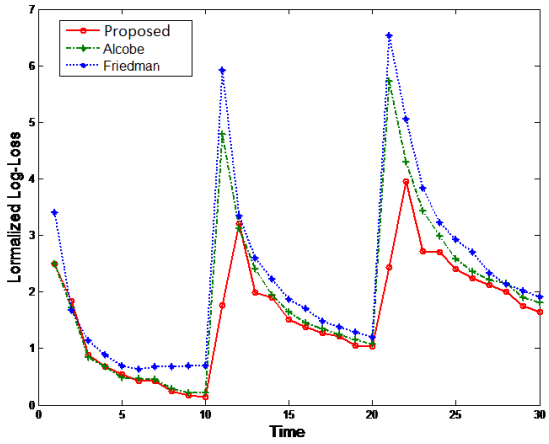

Figure 3. Comparisons of the algorithms on concept drift

In addition, we conduct an experiment on concept drifts. Three experimental networks are used, and the standard Alarm network is used as the first network. The second network is generated by randomly modifying the Alarm network. Then the random modification is performed again to generate the third experimental network. 2000 samples are generated from each of the three networks, totally 6000 samples. The 6000 samples are incrementally inputted to the three algorithms. The experimental results are shown in Fig. 3. It can be seen from the figure that the algorithm can quickly find the moment when the target Bayesian network changes, and can get better learning results than the other two algorithms.

Figure 4 shows the comparison of the running time, it can be seen that the running time of the algorithm is better than that of Alcobe's algorithm and is very close to Fridman's algorithm.

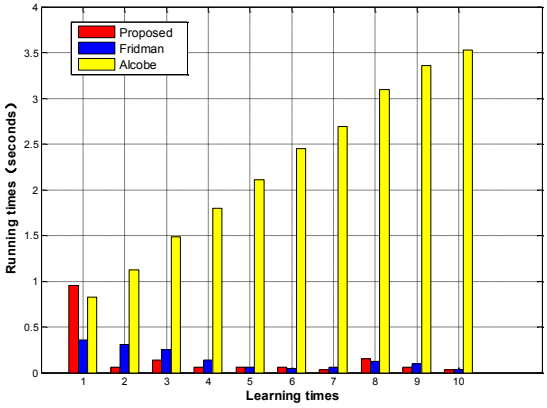

Figure 4. Comparisons of the algorithms on running time

VI. Conclusions and Future Works

In this paper, we proposed a Bayesian network learning algorithm to handle the concept-drift and incremental data. We first design a new incremental scoring function which can adaptively adjust the tendency of matching new and old data in the process of incremental learning. Then we propose an improved greedy-based algorithm to learn the structure of Bayesian network, we introduce tabu-lists for deleting nodes and loop nodes, this strategy can avoid searching for unnecessary redundant nodes and improve learning efficiency. Experimental results show the effectiveness and superior of the method. In future, we will integrate prior knowledge into the learning process based on convex evidence theory [8] to further improve the accuracy of the algorithm. And we also plan to apply the proposed method to performance assessment for electric meter.

References

[1] Yungang Zhu, Dayou Liu, Guifen Chen, Haiyang Jia, Helong Yu. Mathematical Modeling for Active and Dynamic Diagnosis of Crop Diseases based on Bayesian Networks and Incremental Learning. Mathematical and Computer Modelling. 2013, 58(3-4):514-523.

[2] Yungang Zhu, Dayou Liu, Yong Li, Xinhua Wang. Selective and Incremental Fusion for Fuzzy and Uncertain Data based on Probabilistic Graphical Model. Journal of Intelligent and Fuzzy Systems. 2015, 29(6):2397-2403.

[3] Yungang Zhu, Dayou Liu, Haiyang Jia, D. Trinugroho. Incremental Learning of Bayesian Networks based on Chaotic Dual-Population Evolution Strategies and its Application to Nanoelectronics. Journal of Nanoelectronics and Optoelectronics. 2012,7(2):113-118.

[4] Jose A. Gamez, Juan L. Mateo.Jose M. Puerta. Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. Data Mining and Knowledge Discovery. 2011, 22:106-148.

[5] Beinlich I, G Suermondt, R Chavez, G Cooper. The ALRAM monitoring system: A case study with two probabilistic inference. The 2nd European Conference on Artificial Intelligence in Medicine, 1989, 247-256.

[6] N. Friedman, M. Goldszmidt. Sequential update of Bayesian network structure. Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI 1997), Morgan Kaufmann,1997,165-174 .

[7] Josep Roure Alcobe. Incremental methods for Bayesian network structure learning. AI Communications. 2005, 18:61-62.

[8] Dayou Liu, Yungang Zhu, Ni Ni, Jie Liu. Ordered proposition fusion based on consistency and uncertainty measurements. SCIENCE CHINA Information Sciences, 2017, 60(8):082103: 1-19.