Wuhan University
Journal of Natural Sciences

2015, Vol.20 No.3, 214-220

Article ID 1007-1202(2015)03-0214-07
DOI 10.1007/s11859-015-1084-y

Towards Fast and Efficient Algorithm for Learning Bayesian Network

☐ **LI Yanying[1,2], YANG Youlong[1†], ZHU Xiaofeng[1], YANG Wenming[1]**

1 School of Mathematics and Statistics, Xidian University, Xi'an 710071, Shaanxi, China;

2 College of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji 721013, Shaanxi, China

© Wuhan University and Springer-Verlag Berlin Heidelberg 2015

Abstract: Learning Bayesian network structure is one of the most exciting challenges in machine learning. Discovering a correct skeleton of a directed acyclic graph(DAG) is the foundation for dependency analysis algorithms for this problem. Considering the unreliability of high order condition independence(CI) tests, and to improve the efficiency of a dependency analysis algorithm, the key steps are to use few numbers of CI tests and reduce the sizes of conditioning sets as much as possible. Based on these reasons and inspired by the algorithm PC, we present an algorithm, named fast and efficient PC (FEPC), for learning the adjacent neighbourhood of every variable. FEPC implements the CI tests by three kinds of orders, which reduces the high order CI tests significantly. Compared with current algorithm proposals, the experiment results show that FEPC has better accuracy with fewer numbers of condition independence tests and smaller size of conditioning sets. The highest reduction percentage of CI test is 83.3% by EFPC compared with PC algorithm.

Key words: Bayesian network; learning structure; conditional independent test

CLC number: TP 301

Received date: 2014-12-15

Foundation item: Supported by the National Natural Science Foundation of China (61403290,11301408,11401454), the Foundation for Youths of Shaanxi Province (2014JQ1020), the Foundation of Baoji City (2013R7-3) and the Foundation of Baoji University of Arts and Sciences (ZK15081)

Biography: LI Yanying, female, Ph.D. candidate, research direction: Bayesian network. E-mail: liyanying81@163.com

† To whom correspondence should be addressed. E-mail: ylyang@ mail.xidian.edu.cn

0 Introduction

Bayesian networks (BNs) are graphical models that are frequently employed for modelling independencies, conditional independencies and casual relationships in a variety of domains[1]. Learning structures of BNs is significantly important and popular in many domains such as medicine, artificial intelligence and bioinformatics. Particularly, one branch of learning BNs is Bayesian network classifiers[2] which have received more attention in machine learning and data mining in recent years due to the emergence of high dimensional datasets in real-world applications.

Over the last several decades, significant progress has been made for learning BN structures. Nevertheless, the construction of BNs remains a time consuming and intractable task, especially, as the number of variables increase. Generally speaking, BNs structure learning can be broadly categorized into two classes: score+search-based approaches and constraint-based methods. On one hand, score-search approaches define a score function to evaluate the fitness of a network (directed acyclic graph) with respect to the given dataset, which can use the prior probabilities of the structure and parameters. However, score+search-based methods are global learning algorithms which are efficient for learning the full BN structure but incapable of handling large network. On the other hand, constraint-based methods depend on exploring the marginal and conditional independent relationship among all nodes on dataset. We can induce the structure of Bayesian network in agreement with tests results. These tests are often performed by applying statistical or information theoretic measures. The ability to scale up to

hundreds of thousands of variables is a key advantage of constraint-based methods over score-search approaches.

In considerations of convenience and wide scope of practical applications, we are interested in constraint-based methods. Recently, a lot of constraint-based methods have been developed for learning the neighbourhoods or the Markov blankets (MBs) of variables, such as SGS algorithm[3], Inductive Causation[4], PC[3], Three Phase Dependence Analysis (TPDA)[5], KS[6], Grow-shrink (GS)[7], IAMB, interIAMB[8], Fast-IAMB[9], MMPC/MB[10], HITON-PC/MB[11], PCMB[12], IPC-MB[13] and RAI [14]. However, constraint-based methods are plagued by a severe problem: the number of false negatives (missing variables) increases swiftly as the size of the parents and children (PC) sets become large. In the worst case, the PC algorithm requires that the number of condition independence (CI) tests increases exponentially with the number of variables. It is also worthwhile noting that large conditioning sets usually lead to errors for CI tests and result in a poorer estimation of dependent relationships for a small sample size. Thus, we tend to use CI tests of lower orders. Only in this way can we obtain more reliable results of tests. Therefore, it is vital to explore some measures to avoid or alleviate the potential problems of combinatorial explosion for CI test.

In view of above discussions, we propose an accurate and fast local learning algorithm, called FEPC, to learn the PC set of each target node X and then obtain the structure of Bayesian network. Unlike the previous algorithms, the CI test of FEPC is based on the strength of the correlation of variables. For a target variable X, we first test variables with weak correlation with X and take those variables which have strong correlation with X as the conditioning sets. We can use partial correlation analysis, covariance analysis and mutual information and so on to evaluate the strength of correlation between variables. In this paper we use mutual information to sort the variables belonging to candidate PC(X) of the target variable X, and then, based on the sort, use CI tests to remove the (conditional) independent nodes with order. Here we alter the pattern of selecting conditioning sets for each CI test by the sort. Besides, at every iteration of order of CI tests, we alter the sequence of target variables in ascending order according to the number of current neighbours of them. These orderly operations efficiently alleviate the problems discussed above. Simulations illustrated that the proposed algorithms outperform their competitors with better accuracy but fewer numbers of condition independence tests and smaller size of conditioning sets. The reduction percentage can be up to 83.3% in CI tests obtained by the FEPC algorithm compared with the PC algorithm.

The structure of this paper is as follows. Relevant definitions and theorems are given in Section 1. Section 2 introduces the details of our improvement on the PC algorithm for learning the skeleton of DAGs. Section 3 presents some simulation results and compares the improved methods with the closest competitors in details. Final section provides the conclusion.

1 The Basic Concepts and Terms

Some definitions and principles closely related to this paper are introduced in this section. Some well-known definitions and conclusions can refer to papers and books on Bayesian network, e.g. Refs. [1, 3, 4, 9]. We symbolize discrete random variables or nodes in a graph by capital letters. Let upper-case bold-face $\boldsymbol{X}$ mean a set of variables or nodes.

A directed acyclic graph (DAG) $G=(V,E)$ consists of a set of nodes $V=\{X_1,X_2,\cdots,X_n\}$ and a set of directed edges $E\subseteq V\times V$. A Bayesian network (G, $\boldsymbol{P}$) consists of a DAG and a joint probability distribution $\boldsymbol{P}$ on V. The graphical structure-DAG shows the conditional independent relations in a BN qualitatively. The probability distribution $\boldsymbol{P}$ induces the corresponding conditional independent relations quantitatively. X is d-separated from Y given the nodes set $\boldsymbol{Z}$ is denoted by $\mathrm{DSEP}(X;Y\,|\,\boldsymbol{Z})$. Under the distribution, the conditional independence of the variables X and Y given $\boldsymbol{Z}$ is recorded as $\mathrm{Ind}_p(X;Y\,|\,\boldsymbol{Z})$. A directed edge $X_i\rightarrow X_j$ means that $(X_i,X_j)\in E$ and $(X_j,X_i)\notin E$, and X_i is a parent of X_j and X_j is a child of X_i. We indicate a undirected edge with X_i-X_j if both $(X_i,X_j)\in E$ and $(X_j,X_i)\in E$. As a BN satisfies Markov condition, the joint probability distribution on V can be recovered by the following equation:

$$\boldsymbol{P}(X_1,X_2,\cdots,X_n)=\prod \boldsymbol{P}(X_i\,|\,\mathrm{Pa}(X_i))$$

where $\mathrm{Pa}(X_i)$ denotes any combination of the values of the parents of variable X_i. This property gives rise to important savings in storage requirements and also facilitates performances of probability inference. A node X is called a collider if there exist two edges such that $Y\rightarrow X\leftarrow W$. Furthermore, if there is no edge between Y and W, then Y, X and W is called a v-structure in a DAG.

The skeleton of a DAG is the undirected graph which comes from a DAG deleted the direction of edges.

Definition 1 (blocked path) A path between nodes X and Y is blocked by a set of vertices $\boldsymbol{Z}$, if there exists a node W on the path for which one of the following conditions holds:

(i) W is not a collider and $W \in \boldsymbol{Z}$, or

(ii) W is a collider and neither W nor its descendants are in $\boldsymbol{Z}$.

In a DAG, if every path from X to Y is blocked by $\boldsymbol{Z}$, then we said that X and Y are d-separated by set $\boldsymbol{Z}$, and vice versa. We denote it as $\mathrm{Dsep}_G(X;Y \mid \boldsymbol{Z})$. A path between node X and Y is active or open if such W can not be found.

Definition 2 (Faithfulness) A directed acyclic graph G is faithful to a joint probability distribution $\boldsymbol{P}$ over variable set V if and only if every independence present in $\boldsymbol{P}$ is entailed by G and the Markov condition. A distribution $\boldsymbol{P}$ is faithful if and only if there exists a directed acyclic graph G such that G is faithful to $\boldsymbol{P}$ [13].

Theorem 1 Two DAGs are Markov equivalent if and only if they have the same skeleton and the same *v*-structures.

It is well known that we cannot distinguish among the difference of DAGs of a Markov equivalence class. Therefore, in fact, we usually only find a representative of a Markov equivalence class in the structure learning problem.

Theorem 2 A Bayesian network (BN) satisfies the faithfulness condition, then

$$\mathrm{Dsep}_G(X;Y \mid \boldsymbol{Z}) \Leftrightarrow \mathrm{Ind}_P(X;Y \mid \boldsymbol{Z})$$

By this theorem the terms d-separation and probabilistic conditional independence are used interchangeably.

2 Fast and Effective Algorithms: FEPC

A naive strategy for discovering the skeleton would be to check the conditional independence relationships among all nodes given datasets. Thus, we immediately take the well-known PC algorithm into consideration. In the light of complexity and performance of PC, it is fit to exploit sparseness of the graph. In this context, we improve the PC algorithm by adding the additional steps and techniques and thus its speed, efficiency and accuracy are further upgraded. The pseudo codes are listed as follows (Algorithm 1).

Algorithm 1 is to learn the parents and children set of every variable though two phases. The first phase (line 1-5), a sort process, computes the mutual information values between the interesting node X and the nodes in its adjacency set ADJ_X (initial state is all nodes except X), and then rank these neighborhoods in ascending order according to the mutual information values and remove W from the current candidate PC set (ADJ_X) if $I(W,X) \leqslant \varepsilon$, the array is stored in $\mathrm{ADJ\_}X'$. At the same time, a array in inverse sort is stored in $\mathrm{Test\_conditioning\_set}_X$. The sort is vitally important for the next phase. It is well known that larger value of mutual information between X and Y implies that Y is more likely to be the directed neighborhood of X for graphical models. In contrast, smaller mutual information between X and Y means that X is weakly related to Y or Y is marginal or conditional independent of X. Thus the variables in front of $\mathrm{ADJ\_}X'$ are more likely marginal (conditional) independent of X and the variables in front of $\mathrm{Test\_conditioning\_set}_X$ are more likely to be *d*-separate non-adjacent nodes from X.

The second phase, line 8-21, deletes the non-adjacent nodes of each variable by CI tests. At each order of CI tests iteration, we first update the ordering of target variables by step 8 which aims to sort the variables by the sizes of adjacent nodes sets from small to large. It is obvious that the non-adjacent nodes of variable with smaller $\mathrm{ADJ\_}X'$ can be deleted easily and quickly as the conditioning sets won't be too large. Moreover, the Algorithm 1 reduces the non-adjacent nodes of X and Y simultaneously in line 14 and 15. Thus the sizes of $\mathrm{ADJ\_}X'$ of nodes with large adjacent sets will be reduced in advance. Next, we check the independence relationship between target node X and its adjacent variables Y with smaller mutual information values in $\mathrm{ADJ\_}X'$. In this process, we used variable(s) with larger mutual information value(s) with target X from $\mathrm{Test\_conditioning\_set}_X$ as the conditioning sets for CI tests. In other words, we firstly select these nodes sorted at he end of the sequence $\mathrm{ADJ\_}X'$ as the conditioning sets, which is different from the general methods that randomly select node or sets from the candidate PC set. The ordered selection of X, the ordered selection of Y and the ordered selection of the conditioning sets can remove some false positive nodes as early as possible with few number of CI tests. More important, we can reduce the number of high order CI tests and increase the reliability of tests. As we will see the performance of proposed methods in the

following section, the superiority of above steps is shown obviously in experimental section.

Algorithm 1 Fast and efficient PC (FEPC)

Input: complete undirected graph G'; threshold ε; data D
Output: Skeleton
1. $\mathrm{ADJ}_X = U \setminus X$ for each variable X in variable set U
2. for each $X \in U$
3. Initialize $\mathrm{ADJ}_X = \mathrm{ADJ}_X - \{W \mid I(W,X) \leqslant \varepsilon,\ W \in \mathrm{ADJ}_X$
4. Sort the variable $Y \in \mathrm{ADJ}_X$ in ascending order according to the value of mutual information to obtain set ADJ_X' and rank $Y \in \mathrm{ADJ}_X$ in descending order to obtain set $\mathrm{Test\_conditioning\_set}_X$
5. end for
6. $i = 0$
7. while exist some X s.t. $|\mathrm{ADJ\_}X'| > i$
8. Rank those nodes $X \in U$ (if $|\mathrm{ADJ\_}X'| > i$) in ascending order according to the size of ADJ_X' to get set V
9. for every $X \in V$
10. for every $Y \in \mathrm{ADJ\_}X'$
11. Test whether $\exists S \subseteq \mathrm{Test\_conditioning\_set}_X$ with $|S| = i$
12. if $\mathrm{Ind}_P(X \perp Y \mid S)$ holds
13. $\mathrm{ADJ\_}X' = \mathrm{ADJ\_}X' \setminus \{Y\}$ and $\mathrm{ADJ\_}Y' = \mathrm{ADJ\_}Y' \setminus \{X\}$
(keep the sort of the rest variables in and ADJ_Y')
14. $\mathrm{Test\_conditioning\_set}_X = \mathrm{Test\_conditioning\_set}_X \setminus \{Y\}$
And $\mathrm{Test\_conditioning\_set}_Y = \mathrm{Test\_conditioning\_set}_Y \setminus \{X\}$
15. $S_{XY} = S_{XY} \cup S$
16. break
17. end if
18. end for
19. end for
20. $i = i + 1$
21. end while
22.return Skelton

While finding the skeleton as in Algorithm 1, we employ Algorithm 2 to extend the skeleton to a CPDAG, namely, the equivalence class of the underlying DAG. Algorithm 2 outputs a CPDAG, which was proved by Meek in 1995[15].

Algorithm 2 Extending the skeleton to a CPDAG

Input: Skeleton Gskel G_{skel}; separation set S
Output: CPDAG G
1. for all pairs of nonadjacent variables i, j with common neighbour k do
2. if $k \notin S(i, j)$ then
3. Replace $i - k - j$ in G_{skel} by $i \to k \leftarrow j$
4. end if
5. end for
6. In the resulting PDAG, try to orient as many undirected edges as possible by repeated application of the following three rules.
7. Orient $j - k$ into $j \to k$ whenever there is an arrow $i \to j$ such that i and k are nonadjacent. (R1)
8. Orient $i - j$ into $i \to j$ whenever there exists a chain $i \to k \to j$ (R2)
9. Orient $i - j$ into $i \to j$ whenever there exist two chains $i - k \to j$ and $i - l \to j$ such that k and l are nonadjacent. (R3)

In order to illustrate the working mechanism of these three ordered selections of FEPC, we introduce a concrete example. Suppose we have a dataset sampled from an underlying Bayesian network as Fig. 1. And the CI tests are reliable. Algorithm 1 firstly ranks target variables, without loss of generality, takes A and F for example, and supposes that $\{H, D, A\}$ and $\{B, C, E, D, F\}$ are the neighbor sets of F and A after the i-th loop, respectively. Because the neighbor size of F is smaller than that of A, we first check F according to line 8, then we employ two times CI tests of order 1 at most to discover the conditional independent relationship between A and F. On the contrary, for A, we may get the same result by operating four tests. Next, we explain the working mechanism of line 9-19. Let the current target variable be A, our task is to find out the set of the parents and children of A. Suppose that

$$I(A,D) > I(A,B) > I(A,C) > I(A,E) > I(A,G) > I(A,F) > I(A,H) = 0$$

then we remove H from ADJ_A after line 4 of Algorithm 1, and we have

$$\mathrm{ADJ\_}X' = \{F, G, E, C, B, D\}$$
$$\mathrm{Test\_conditioning\_set}_A = \{D, B, C, E, G, F\}$$

separately. Consecutively, we determine conditional independent relation between target variable A and vari-

ables belonging to ADJ_X' in sequence. For example, we operate one order CI test for A and F, then $\{D\},\{B\},\{C\},\{E\},\{G\}$ will be used as conditioning set one after another instead of randomly selecting from Test_conditioning_set$_A \setminus F$. As $\mathrm{Dsep}_G(A;F \mid D)$ and reliability of CI tests, the current loop will be broken immediately. That is what other tests can be effectively avoided. Similarly, other nodes individually repeat this procedure until |Test_conditioning_set|＜sepsetsize or i. However, if by other general algorithms and in worse case, we just first test variable D with the maximum association with A, in this context, we may have to traverse ergodic all nodes belonging to Cand_PC(A)$\setminus F$ or their combinations sometimes and operate unnecessary CI tests and then high order tests maybe occur in this process. Moreover, these high order tests may low the learning accuracy.

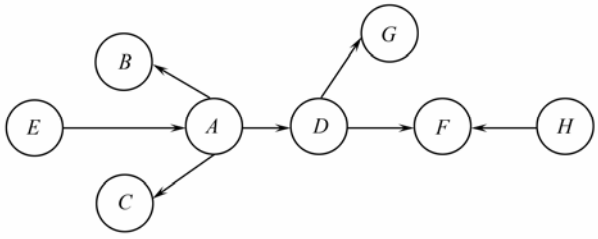

Fig. 1 An example of Bayesian network

3 Experiments and Analysis

In this section, we compare the FEPC algorithm with other state-of-the-art and prototypical algorithms which deal with the same problems respectively. The experiments are run on a Pentium3.19 GHz with 1.96 GB RAM using Windows XP system and Matlab inversion R2009a.

We compare our extend method with the typical algorithms PC, SC, TPDA, MMHC and RAI. We test all algorithms on two well-known networks. The first is Alarm network[16] which is a widely accepted benchmark for evaluating the performance of many algorithms. The second is Insurance network[17] which is for estimating the expected claim costs for a car insurance policyholder, consists of 27 vertices. To guarantee the reliability of the experimental results and fairness for comparison, we use 10000 samples which are randomly generated by the true networks, respectively. Meanwhile, the number of data sets with the same network are 10. Notice that, the significance level α for the conditional independence test is set to 0.05 and threshold $\varepsilon = 0.001$ for all algorithms used in this paper. Besides, our implementation is based on the Bayesian network toolbox written by Murphy[18].

The accuracy of a learned structure can be measured using several scores and criterions. However, some of the scores suggested in the literatures are not always accurate or related to the true structure. Thus we must select proper metrics. Following Spirtes *et al*[3] and Tsamardinos *et al*[17], we use five types of structural errors to evaluate the accuracy of our algorithm. An extra edge (EE) error is an edge learned by the employed algorithm, nevertheless, the learned edge is not covered in the true network. A missing edge (ME) error is due to an edge missed by the algorithm although it is contained in the true graph. An extra direction (ED) error is the direction of an edge, which appears in the learned network but not in the true graph, conversely, a missing direction (MD) error implies that there is an edge direction in the true graph rather than in the learned graph. Last but not least, a reversed direction (RD) error refers to an edge direction in the learned network that is opposite to the edge direction in the true graph.

Table 1 lists the average results of the five structural errors for each algorithm in 10 independent runs for Alarm network. The table also depicts the total directional error DE, which is the sum of ED, MD and RD. In a similar way, SHD sums all five structural errors. Moreover, Table 1 illustrates that the lowest DE and EE errors are obtained by FEPC and the lowest ME error is achieved by MMHC. The last column describes the overwhelming advantage of FEPC over all other algorithms by the SHD errors.

Table 1 Structural errors of algorithms as averaged on 10 independent runs for Alarm network

Algorithm	ED	MD	RD	DE	EE	ME	SHD
SC	2.7	1.3	12.3	16.3	7.0	3.3	26.6
MMHC	4.0	0.0	10.3	14.3	2.3	0.3	17.0
PC	1.0	4.0	0.0	5.0	0.0	9.0	14.0
TPDA	1.0	4.2	0.0	5.2	2.0	3.6	10.8
RAI	3.3	1.0	0.0	4.3	2.6	2.3	9.2
FEPC	1.3	1.0	0.0	2.3	0.0	5.0	7.3

Table 2 summarizes the simulation results of the five structural errors for each algorithm in 10 independent runs for Insurance network. The values of the DE and SHD errors in Table 2 illustrate that FEPC is significantly superior to all other algorithms.

The complexity of FEPC was evaluated in comparison to that of the PC algorithm by the number of CI

Table 2 Structural errors of algorithms as averaged on 10 independent runs for Insurance network

Algorithm	ED	MD	RD	DE	EE	ME	SHD
SC	10.8	4.1	15.4	30.3	1.8	9.7	43.6
MMHC	4.0	0.0	14.6	18.6	1.2	9.4	29.2
PC	6.3	3.3	1.7	11.3	0.3	9.0	20.6
TPDA	8.3	2.0	13.1	23.1	2.4	12.0	37.5
RAI	0.4	0.0	13.4	13.8	1.2	6.7	21.7
FEPC	6.3	3.3	1.3	10.9	0.2	9.0	20.2

tests required to learn these two networks. We assume that CI tests are reliable and compare the number of CI tests. The average number of CI tests required by each algorithm is shown in Fig. 2(a) and Fig. 3(a) for increasing the size of conditioning set for CI tests on Alarm network and Insurance network, respectively. Fig.2(b) and Fig.3(b) demonstrate the percentage of the number of CI tests saved by FEPC compared with PC for increasing orders on Alarm network and Insurance network sepa

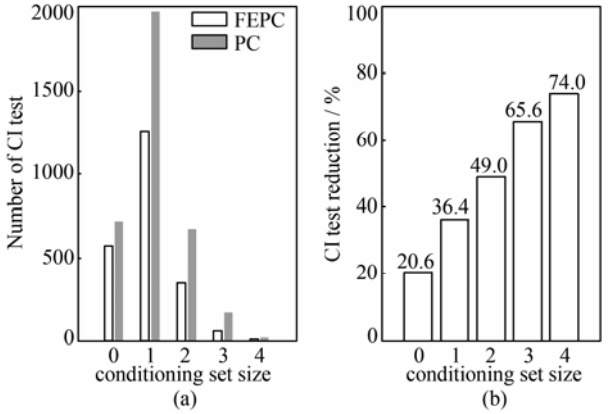

Fig. 2 Experimental results on Alarm network

(a) The average number of CI tests required by FEPC and PC for learning Alarm network, respectively; (b) Reduction percentage in CI tests obtained by the FEPC algorithm compared with the PC algorithm

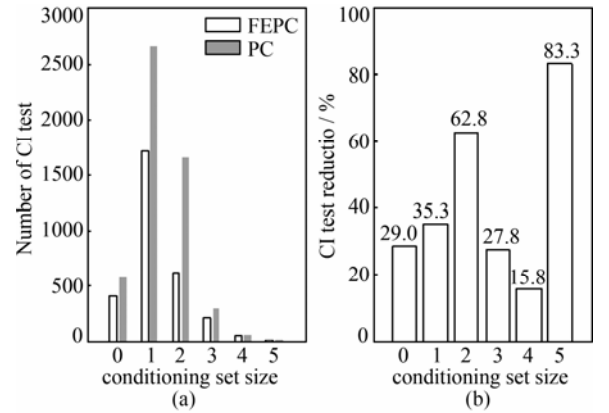

Fig. 3 Experimental results on Insurance network

(a) The average number of CI tests required by FEPC and PC for learning Insurance network, respectively; (b) Reduction percentage in CI tests obtained by the FEPC algorithm compared with the PC algorithm

rately. The highest reduction percentages of CI test are 83.3% on Insurance network and 74% on Alarm network by FEPC compared with PC algorithm, respectively. Obviously, the FEPC algorithm reduce the number of CI tests of each order, moreover, preponderance of FEPC over the PC algorithm is more outstanding for high orders.

4 Conclusion

In this paper, we proposed algorithm FEPC which successfully avoids CI tests with large conditioning sets and uses as fewer CI tests as possible. Here using mutual information to sort the nodes in the candidate PC set of target variables X is the key step for FEPC. By the sort, we can not only select the possible non-adjacent nodes Y in a specific sort, but also select the conditioning sets for Y and X in a fixed sort. These non-random selection can efficiently reduce the orders CI tests and the number of high order of CI tests and remove false positive nodes as early as possible.

In addition, we compared the proposed algorithm with other state-of-the-art algorithms on some standard networks. Simulations results demonstrate that the proposed method outperforms its competitive algorithms with respect to accuracy and complexity.

We plan to extend our simulation experiment on large network and study the ability of handling large networks of FEPC. Furthermore, although we assume in this paper that the data are completely observed, however in practice, missing data or data with latent variables may arise[19,20], so we try to generalize the proposed algorithms to missing data or data with latent variables in the next research.

References

[1] Yang Y, Wu Y. On the properties of concept classes induced by some multiple-valued Bayesian networks [J]. *Information Sciences*, 2012, **184**: 155-165.

[2] Chang H, Lee W. An information theoretic filter method for feature weighting in naive Bayes [J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2014, **28**(5): 1451007.

[3] Spirtes P, Glymour C N, Scheines R. *Causation*, *Prediction and Search* [M]. Cambridge: MIT Press, 2000.

[4] Pearl J. *Causality*: *Models*, *Reasoning and Inference*[M]. Cambridge: MIT Press, 2000.

[5] Cheng J, David B, Liu W. Learning Bayesian networks from data: An efficient approach based on information theory [C]

//*Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science*. Berlin: Springer-Verlag, 2005, **3518**: 474-479.

[6] Koller D, Sahami M. Toward optimal feature selection[C] // *Proc of International Conference on Machine Learning*. San Francisco: Morgan Kaufmann Publishers, 1996: 284-292.

[7] Margaritis D, Thrun S. Bayesian network induction via local neighborhoods[C]//*Advances in Neural Information Processing Systems*. Cambridge: MIT Press, 1999.

[8] Tsamardinos I, Aliferis C F. Towards principled feature selection: Relevancy, filters and wrappers[C] // *Proc* 9*th International Workshop on Artificial Intelligence and Statistics*. San Francisco: Morgan Kaufmann Publishers, 2003.

[9] Yaramakala S, Margaritis D. Speculative Markov blanket discovery for optimal feature selection, data mining[C] // 5*th IEEE International Conference on IEEE*, 5*th IEEE International Conference on Data Mining*. Washington D C: IEEE Press, 2005: 809-812.

[10] Tsamardinos I, Aliferis C F, Statnikov A. Time and sample efficient discovery of Markov blankets and direct causal relations[C]//*Proc* 9*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York：ACM Press, 2003: 673-678.

[11] Aliferis C F, Tsamardinos I, Statnikov A. HITON: A novel Markov blanket algorithm for optimal variable selection[C] //*AMIA Annual Symposium Proceedings*. Maryland: American Medical Informatics Association, 2003: 21-25.

[12] Pena J M, Nilsson R, Bjorkegren J. Towards scalable and data efficient learning of Markov boundaries [J]. *International Journal of Approximate Reasoning*, 2007, **45**: 211-232.

[13] Fu S, Desmarais M C. Fast Markov blanket discovery algorithm via local learning within single pass [C] // *Advances in Artificial Intelligence*. New York: Springer-Verlag, 2008: 96-107.

[14] Yehezkel R, Lerner B. Bayesian network structure learning by recursive autonomy identification[J]. *Journal of Machine Learning Research*, 2009, **10**: 1527-1570.

[15] Meek C. Causal inference and causal explanation with background knowledge[C]//*Proceedings of the* 11*th Conference on Uncertainty in Artifical Intelligence*. San Francisco: Morgan Kaufmann Publishers, 1995: 403-410.

[16] Heckerman D, Geiger D, Chickering D M. Learning Bayesian networks: The combination of knowledge and statistical data [J]. *Machine Learning*, 1995, **20**(3): 197-243.

[17] Tsamardinos I, Brown L E, Aliferic C F. The Max-minhill-climbing Bayesian network structure learning algorithm [J]. *Machine Learning*, 2006, **65**(1): 31-78.

[18] Murphy K. The Bayes net toolbox for Matlab[J]. *Computing Science and Statistics*,2010, **33**: 1024-1034.

[19] Colombo D, Maathuis1 M H, Kalisch M, *et al*. Learning high-dimensional directed acyclic graphs withlatent and selection variables [J]. *The Annals of Statistics*, 2012, **40**(1): 294-321.

[20] Liu X, Yang Y, Zhu M. Structure learning of causal bayesian networks based on adjacent nodes [J]. *International Journal on Artificial Intelligence Tools*(*IJAIT*), 2013, **22**(2): 1-18.

□