Improved Bayesian Network Structure Learning with Node Ordering via K2 Algorithm*

Zhongqiang Wei, Hongzhe Xu, Wen Li, Xiaolin Gui, and Xiaozhou Wu

Shaanxi Key Lab of Computer Network, Xi'an Jiaotong University, Xi'an, China
{xuhz,leewhen,xlgui}@mail.xjtu.edu.cn

Abstract. The precise construction of Bayesian network classifier from database is an NP-hard problem and still one of the most exciting challenges. K2 algorithm can reduce search space effectively, improve learning efficiency, but it requires the initial node ordering as input, which is very limited by the absence of the priori information. On the other hand, search process of K2 algorithm uses a greedy search strategy and solutions are easy to fall into local optimization. In this paper, we present an improved Bayesian network structure learning with node ordering via K2 algorithm. This algorithm generates an effective node ordering as input based on conditional mutual information. The K2 algorithm is also improved combining with Simulated Annealing algorithm in order to avoid falling into the local optimization. Experimental results over two benchmark networks Asia and Alarm show that this new improved algorithm has higher classification accuracy and better degree of data matching.

Keywords: Bayesian Network Classifier, Structure Learning, Search Strategy, Conditional Mutual Information, K2 Algorithm.

1 Introduction

Pearl [1] put forward Bayesian network in 1980s, after that Bayesian network was widely applied to the classification process of data mining. The process of constructing Bayesian network classifier can be generally divided into three steps: (1)Learning Bayesian network topological structure from the training data which represents the dependent relationships between variables. (2)Learning Conditional Probability Table based on network structure. (3)Classifying and making decisions through finding the maximum posteriori probability. Among these three steps, Bayesian network structure learning is the key to the construction of Bayesian network model.

Bayesian network structure learning can be divided into two main categories: learning algorithms based-on dependent relationships [2, 3] and learning algorithms based-on scoring functions [4-6]. Learning algorithms based-on dependency analysis construct Bayesian network structure through the establishment of the conditional

* This research is supported by the central university basic scientific research business (XKJC2014008).

D.-S. Huang et al. (Eds.): ICIC 2014, LNAI 8589, pp. 44–55, 2014.

dependent relationships between nodes. The learning process of this kind of algorithm is more intuitive, conditional independence test and the search of network structure can be separated, but these algorithms are oversensitive with the error of condition independence test. Learning algorithms based-on scoring functions view learning as the optimization process, and select the best network structure which maximizes the scoring value. Generally network structures learned by this kind of algorithms have high precision, but structures are inclined to fall into local optimum.

K2 algorithm is a classical Bayesian network structure learning algorithm based-on scoring search proposed by Cooper and Herskovits [6], which combines Bayesian scoring method and hill-climbing search strategy. When using K2 algorithm, it has to determine the initial node ordering and maximal number of parents of each node, which are hard to obtain in many practical applications. In order to solve these problems, researchers have put forward many improved algorithms [7-9], the application effects of these algorithms are not very ideal. K2 algorithm is computationally efficient given an initial node ordering, which reduces the search space significantly. But, an improper node ordering may give poor results. So, it is of great importance to provide the K2 algorithm with an effective node ordering. David et al. [10] introduced polynomial algorithms for finding the highest scoring network structures in the special case where every node has at most k=1 parent. These kinds of algorithms limit parent number of each to great extent, consequently resulting network structures are close to a maximum spanning tree algorithm, which reflects the relationships between the nodes not well. Chen et al. [3] applied mutual information to determine node ordering and K2 algorithm to search node ordering space to learn network structure. But this algorithm didn't use class attribute information to obtain initial node ordering, which play an important role in node ordering. Simulated Annealing (SA) [11] algorithm is a search-score learning algorithm based-on the Monte Carlo approach. It has better global optimization ability, but its efficiency is low. Combining K2 algorithm and Simulated Annealing algorithm not only can be avoidance of the local optimum, but can also improve learning efficiency. In those algorithms based-on scoring search, researchers have proposed many scoring methods [12-15]. In this paper, we use the likelihood-equivalence Bayesian Dirichlet score metric [12] as the evaluation standard of network structures.

In this paper, the method of an effective node ordering is put forward based on conditional mutual information (CMI). Then an improved K2 algorithm called K2-SA algorithm is used to learn Bayesian Network structure and a heuristic Bayesian Network classification algorithm is proposed. Finally experimental results show the effectiveness of this algorithm.

This paper is organized as follows. Section 2 explains how to generate the initial node ordering based on conditional mutual information in detail. The improved algorithm K2-SA is given in Section 3. Experimental results of two benchmark network data sets with known structure are presented in Section 4. Finally, conclusion of this paper is given.

2 An Effective Node Ordering Based-On Conditional Mutual Information

Learning a Bayesian network structure is a time-consuming task. When providing an initial node ordering, K2 algorithm can improve learning efficiency by reducing search space. However, the performance of K2 algorithm is greatly affected by the initial node ordering. This paper presents a method to obtain an effective node ordering. It consists of three main components. Firstly, obtain a weighted undirected graph structure based on conditional mutual information. Secondly, assign orientations for every edge of the undirected structure. Thirdly, establish a maximal weight spanning tree (MWST) based on the directed graph.

2.1 Obtain a Weighted Undirected Graph Structure Based on CMI

Firstly, we obtain an undirected network structure based on conditional mutual information and the connectivity is established in the network. The values of CMI among pairs of random variables are assigned to edges in the network as the weight. Then a weighted undirected network structure can be obtained.

The degree of dependence between two random variables can be acquired by evaluating CMI value between these two variables. The high CMI value between variables X_i and $X_j (1 \le i, j \le n, n$ is the number of random variables) given variable Z represents variable X_i and X_j have a great degree of mutual dependence under the condition of giving variable Z. This dependence can be used to establish connections between variables.

The CMI between two random variables X_i and X_j given random variable Z, denoted by $I(X_i; X_j \mid Z)$, is mathematically defined as follows:

$$I(X_i; X_j \mid Z) = \sum_{k}^{q_z} \sum_{l}^{q_l} \sum_{m}^{q_j} P(x_l, y_m \mid z_k) \log \frac{P(x_l, y_m \mid z_k)}{P(x_l \mid z_k) P(y_m \mid z_k)}, i \ne j \tag{1}$$

where q_z represents the number of all the possible values of variable Z, z_k represents all the possible values of variable Z, q_i represents the number of all the possible values of variable X_i, x_l represents all the possible values of variable X_i, q_j represents the number of all the possible values of variable X_j, y_m represents all the possible values of variable X_j, $i \ne j$ represents random variables X_i and X_j cannot be the same variable, both variable X_i and X_j are different from variable Z. Importantly, the CMI is symmetric in nature, that is $I(X_i; X_j \mid Z) = I(X_j; X_i \mid Z)$.

To obtain an undirected graph structure, the CMI is computed between every two node pairs for all the nodes in the network we except to construct. Assuming that a given variable set $X = \{X_1, X_2, ..., X_n\}$ which has n variables or attributes, and a class attribute C. Then the CMI value between any two variables X_i and X_j can be computed as $I(X_i; X_j \mid C), 1 \le i, j \le n, i \ne j$. An edge is added between these two

nodes or variables and the weight is assigned to $I(X_i; X_j \mid C)$. Because of the symmetric nature of CMI, the edges established in the graph structure are undirected. Compute CMI for each node or variable pairs, and add the undirected edges and the corresponding weights. After that, a weighted undirected complete graph structure can be obtained.

2.2 Assign Orientations for Every Edge of the Undirected Structure

This phase is to assign orientations for every edge of the undirected graph structure in order to obtain the initial node ordering information. The conditional relative average entropy (CRAE) [16] is used to determine orientations for edges. For two random variables X_i and X_j, their CRAE is defined as:

$$CRAE(X_j \to X_i) = \frac{H(X_i \mid X_j)}{H(X_i) \cdot \mid X_i \mid}, i \neq j \tag{2}$$

where $H(X_i) = -\sum_{x_l} P(x_l) \log P(x_l))$ is the entropy of random variable X_i, $H(X_i \mid X_j) = -\sum_{x_l, x_m} P(x_l, x_m) \log P(x_l \mid x_m)$ is the conditional entropy of random variable X_i given random variable X_j, $\mid X_i \mid$ represents the number of all possible values of X_i, x_l represents all possible values of X_i, x_m represents all possible values of X_j.

For two nodes X_i and X_j, if $CRAE(X_j \to X_i) > CRAE(X_i \to X_j)$ the orientation of edge between nodes X_i and X_j is $X_j \to X_i$. If $CRAE(X_i \to X_j) > CRAE(X_j \to X_i)$, the edge orientation between nodes X_i and X_j is $X_i \to X_j$. If $CRAE(X_i \to X_j) = CRAE(X_j \to X_i)$, the edge between those two nodes is not assigned the orientation at this point. On this occasion, we will use Bayesian score method to set the orientations for the remaining undirected edges after this step.

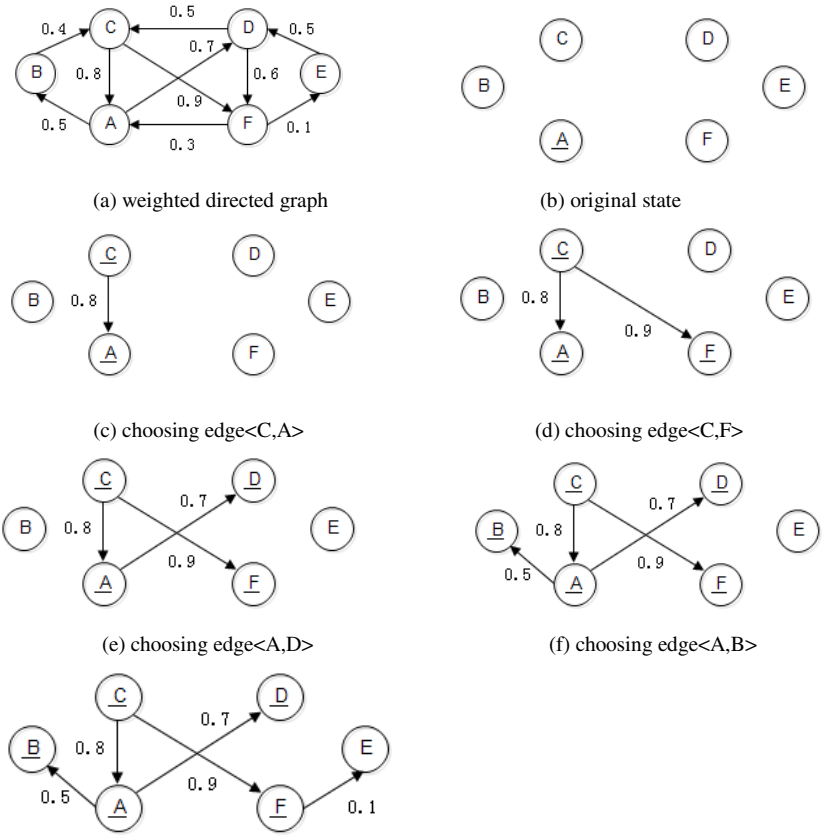
By using CRAE, most edges can be set the orientations and the number of edges whose orientations are remaining to be assigned is small. Thus, the exhaustive search combining with Bayesian score method is used to determine the remaining undirected edges' orientations.

After the above steps, a weighted directed graph structure can be acquired. In the next section, this directed graph is used to establish a maximal weight spanning tree (MWST) in order to obtain the initial node ordering.

2.3 Establish a MWST Based on the Weighted Directed Graph

This phase is to construct the MWST of the weighted directed graph. Assuming that the initial graph is $G(V, E)$ and the generated MWST is $T(U, D)$, the process is as follows:

- Initialize U and D as the empty set, that is $U \leftarrow \varnothing, D \leftarrow \varnothing$.
- Select a node u_0 randomly from V in the graph G and put this node into G. Then $U = \{u_0\}$ and $V = V - \{u_0\}$. A directed edge $< u_0, v >$ or $< v, u_0 >$ that relates to node u_0 and has maximal weight is selected. The node v is added into U, then $U = U \cup \{v\}$ and $V = V - \{v\}$. This directed edge is added into D.
- Select a directed edge $< x, y >$ in the E from graph G, which satisfies the following conditions: 1) one of the two nodes associated with this edge is in V, the other one is in U; 2) the in-degree of the arc head of this edge is 0; 3) this edge has the maximal weight among all the directed edges which satisfy condition 1) and 2). The selected edge $< x, y >$ is added into D and the node that associates with this edge and is absent in U is added into U.
- Repeat the third step until U contains all the nodes.



(a) weighted directed graph

(b) original state

(c) choosing edge<C,A>

(d) choosing edge<C,F>

(e) choosing edge<A,D>

(f) choosing edge<A,B>

(g) choosing edge<F,E>

Fig. 1. Process of construct the MWST of the weighted directed graph

After the above steps, a MWST based on the directed graph can be obtained. The initial node ordering can be determined according to this MWST. As shown in Fig.1, node A is selected as initial node randomly, and weights of directed edges are set as (a) in Fig.1. The resulting MWST is shown as Fig.1.(g), which the initial node ordering can be obtained according to it. For example, a topological sort algorithm can be used to obtain node ordering, which will generate many optional node ordering plans. Sorting directed edges according to weight, node ordering can be determined more accurate. In Fig.1, the node ordering is unique, that is CFABDE. This obtained node ordering will be used to learn the Bayesian network structure effectively in order to construct Bayesian network model.

3 K2-SA Algorithm

K2 algorithm with greedy search strategy can reduce the solution space significantly, but it often can't get the global optimal solution. SA algorithm is a random optimization algorithm, using the Metropolis criterion with probabilistic jumping properties to find the global optimal solution of the objective function randomly in the solution space. SA algorithm utilizes the similarity between annealing process and combination optimal problem to simulate the energy of physical system into the optimization problem of objective function, thereby search the global optimal value of optimization problem. The characteristic of SA algorithm can remedy the insufficiency of greedy search strategy that falls into local optimal solution easily. The algorithm combining K2 and SA algorithm (K2-SA) controls solving process to the optimization direction of minimum, and can escape from local extreme points with accepting inferior solutions at a certain probability. As long as the initial temperature is high enough and the annealing process is slow enough, this algorithm can converge to the global optimal solution. Therefore, K2 and SA algorithm can be combined to learn more optimal network structure.

When giving the node ordering, K2-SA algorithm searches the set of parents for each node orderly in order to establish the network structure. For any node X_i, only nodes appearing before node X_i in the node ordering can be the parents of node X_i. For example, node X_j comes prior to node X_i in the ordering, then node X_j can be a parent of node X_i, but node X_i cannot be a parent of node X_j.

The process of using K2-SA algorithm to choose optimal set of parents for each node is as following:

- Initialize the parent node set $Pa(X_i)$ of node X_i an empty set, which is $Pa(X_i) \leftarrow \varnothing$.
- Calculate evaluation function:

$$f(S) = CH(< X_i, Pa(X_i) >| D) = \sum_{j=1}^{q_i} [\log \frac{\Gamma(\alpha_{ijk})}{\Gamma(\alpha_{ij} + N_{ij})} + \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}] \quad (3)$$

- where q_i represents all the configuration number of parents of node X_i, N_{ijk} is the sample number of the i th variable when the configuration of its parents is the j th and its value is the k th, N_{ij} is the sample of the i th variable when the configuration of its parents is the j th, that is $N_{ij} = \sum_k N_{ijk}$, α_{ijk} represents the hyper-parameter in the prior distribution.
- Pick the score function value minus and convert it to energy representation of simulated annealing, that is $E_{old} = -f(S)$. E_{old} represents the energy of simulated annealing, and lower energy value has higher probability of acceptance.
- Set initial temperature T_0, let current temperature $T_k = T_0$, and set minimum temperature T_g.
- Set the iteration counter $l = 0$ and the termination condition of iteration is $l_e = L(x)$, where $L(x)$ is variable step function.
- Under the current temperature T_k, perform the following actions:

— Step1: Sequentially select all nodes which are before node X_i and don't exist in the parent node set $Pa(X_i)$, calculate the scoring value for each such node, add the node which makes the scoring value maximal into the parent node set of node X_i, then produce a new adjacent set $Pa_{new}(X_i)$.

— Step2: Calculate the value of evaluation function $f(S')$. Pick minus the score function value and convert it to energy representation of simulated annealing, that is $E_{new} = -f(S')$.

— Step3: If $Pa(X_i)$ is an empty set, the new node set is accepted as the parent node set of node X_i. If not, the value of E_{old} and E_{new} will be compared. If $E_{new} < E_{old}$, the new parent node set will be accepted, that is $Pa(X_i) = Pa_{new}(X_i)$, $E_{new} = E_{old}$. If $E_{new} \geq E_{old}$, the new parent node will be accepted with probability P.

— Step4: If accepting the new set of parents, the iteration counter l is set to 0 and go to Step1. If not, set $l = l + 1$ and go to the next step.

- Judge whether the iteration number l achieves the maximum iteration number or not. If $l < l_g$, go to the sixth step and continue to iterate. If not, algorithm has reached balance state under the current temperature and executes the next step.
- Determine whether current temperature achieves the minimum temperature or not. If $T_k > T_g$, carry on temperature decrease according to formula $T_{k+1} = r \cdot T_k$. The step size under each temperature is $L(x) = c \cdot C \cdot r \cdot P_{acc}$, where c is a constant, C is dimension of node states, r is decline coefficient, P_{acc} is the probability of acceptance. Then go to the fifth step. If not, continue to execute the next step.
- Output the parent node set $Pa(X_i)$.

With respect to K2-SA algorithm, there are some descriptions that need to explain:

- If initial temperature is set high enough, almost all of the adjacent can be accepted, and the algorithm can converge to global optimum. However, the resulting computational efficiency will be lower. Therefore, in the real case, the setting of initial temperature generally adopt a compromise with considering the quality and efficiency of optimization. In this paper, we produce several states (sets of parents of each node) firstly. Secondly, the difference values $|\Delta|$ of score functions between every two states are computed and select the maximal difference value as $|\Delta_{\max}|$. Then the initial temperature T_0 is determined as $T_0 = -|\Delta_{\max}| / \ln P_0$ where P_0 is the initial acceptance probability that needs to set manually.
- If algorithm has reached balance state under the current temperature, annealing action will be taken. The method used in this paper is $T_{k+1} = r \cdot T_k$ where $0.5 < r < 0.99$ represents annealing coefficient.
- The acceptance criterion for new solutions is the main difference between SA algorithm and the general search algorithm. Method of energy value computed in this paper is picking the Bayesian score function value minus, which is $E = -f(S')$. The specific approach of accepting the new state with probability P is that algorithm produces a random number γ between 0 and 1. If $P > \gamma$, $Pa_{new}(X_i)$ will be accepted as the new parent node set of node X_i and E_{old} is set as E_{new}. If not, keep the original state and parent node set unchanged.
- Generally, the termination condition is that several successive solutions are not accepted, at this time, the isothermal search at this temperature will be terminated. In this paper, a dynamic control method of iterations is employed and the step size is represented as $L(x) = c \cdot C \cdot r \cdot P_{acc}$. This control mode of iterations can guarantee to achieve the balance state at current temperature, reduce the redundant iterative number, and improve the speed of operation to a certain extent.
- K2 algorithm predefines the maximum number of parent nodes. When existing a node whose parent nodes exceed the maximum in the network structure, it needs to correct that parent node in order to make the parent node number of the current node no more than the maximum. The correction algorithm uses the mutual information between nodes and its parent nodes. Select a parent node which has minimum mutual information to delete sequentially until the condition of upper limit of number of parents is satisfied. The correction algorithm selects several nodes from the parent node set to delete so that the criterion about the maximum number of parent nodes can be satisfied.

According to the above method, we can obtain near optimal set of parents for each node. Bayesian network structure can be constructed by using these sets of parents. After learning conditional probability tables of network structure, Bayesian network model can be establish.

4 Experiment and Analysis

In order to validate the method proposed by this paper, experimental procedure uses 10-fold cross-validation method. Assume that all attributes are discrete and there is no missing value. Experiment tests one hundred times on each data set, selects the best result and average result respectively and calculates the standard deviation of results. Before experimenting, data need to be discretized. This paper utilizes conditional mutual information to determine the initial node ordering. Then K2-SA algorithm is used to learn Bayesian Network structure and conditional probabilities are also learned. Finally, the Bayesian Network classifier is constructed. All parameters in the experiment employ BDe [12] priori information and the global union tree algorithm is employed as inference algorithm. The priori value α_{ijk} is set as 1. The initial temperature of simulated annealing is set as $T_0 = 1$, the termination temperature is set as $T_g = 0.001$, the jumping probability P is set as 0.2 and the temperature decline coefficient is set as $r = 0.8$.

Parameters used for analysis and comparisons among several algorithms are given as follows:

1. **S (Score):** The BDe score value of the best individual network learned.
2. **CA (Classification Accuracy):** The value of classification accuracy of the best individual network learned.
3. **MR (Mean Result):** This is the averaged result over 100 runs of several algorithms on 100 different and independent datasets for network structures.
4. **SD (Standard Deviation):** It is the Standard Deviation of the results.
5. **BR (Best Result):** It is the best result.

Experimental results are carried with our method on two standard network data sets (Asia, Alarm). Other algorithms (TAN, Hill Climbing, and Random K2) are also performed to make a comparative study.

Asia [17]: The Asia network is a small Bayesian Network, and is used for a fictitious medical example. This example researches whether a patient has tuberculosis, lung cancer, or bronchitis, related to the chest clinic. It consists of eight nodes with eight edges connecting them. Each random variable (node) is discrete in nature and takes two discrete states (values).

Alarm [18]: The Alarm network is a medical diagnostic system for patient monitoring. This network consists of 37 with 46 edges connecting them. The random variables in the network are discrete in nature and can take two, three or four states.

Table 1. Experimental result comparison of network structure learned by these methods (Asia, Alarm)

Methods	MR		SD		BR	
	Asia	*Alarm*	*Asia*	*Alarm*	*Asia*	*Alarm*
Our method	97.55	97.54	16.4	17.4	98.51	98.65
TAN	93.64	94.77	15.8	16.1	94.98	95.73
Hill Climbing	97.42	97.55	19.2	19.0	98.21	98.60
Random K2	94.32	94.26	22.3	21.1	96.39	96.44

As can be seen from Table 1, both the presented method and other three algorithms have high classification accuracy, but classification accuracy of our method is higher than other three classification algorithms as a whole. In order to compare network structure performance of these methods, this paper compares network structures learned by these methods. Table 2 and 3 list the BDe score values of network structures generated by these methods.

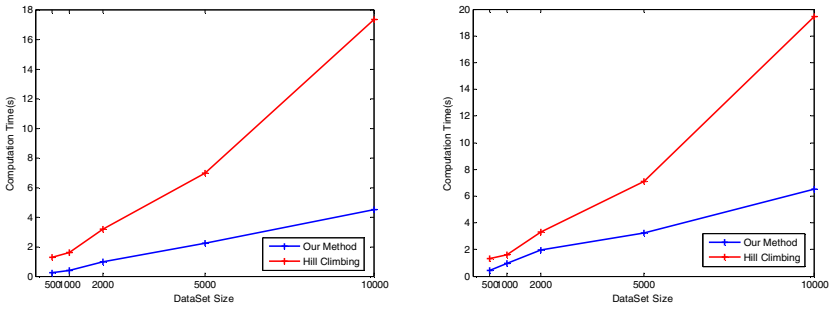
Table 2. Score value comparison of network structures learned by these methods (Asia)

Methods	MR	SD	BR
Our method	-525.10	17.5	-523.21
TAN	-530.20	17.4	-526.43
Hill Climbing	-531.17	19.2	-526.33
Random K2	-529.84	22.7	-523.97

Table 3. Score value comparison of network structures learned by these methods (Alarm)

Methods	MR	SD	BR
Our method	-526.17	18.6	-524.22
TAN	-555.26	17.8	-553.74
Hill Climbing	-529.63	20.1	-525.72
Random K2	-532.38	22.6	-526.13

As can be seen from data in Table 2 and 3, BDe score value of our method is larger than compared with other three classification algorithms, which shows that the network structure learned by our method is more in line with real variable dependent relationships and has a higher degree of data fitting.



(a) Asia Network (b) Alarm Network

Fig. 2. Computation time of learning Alarm network (seconds)

As can be seen from the above three forms, results gained by our method is similar to Hill Climbing algorithm. However, Fig.2 shows that the growth rate of computation time (seconds) of Hill Climbing is much higher than our method with the growth of dataset size.

Experimental results show that the novel method proposed by this paper has good classification performance and the network structure learned is more reasonable compared with the three algorithms. This network structure fits data better and can be more truly reflect the dependencies between variables.

5 Conclusion

In view of K2 algorithm needs to determine the initial node sequence and is easy to fall into local optimum, this paper presents a novel method which uses conditional mutual information to generate the initial node ordering and employs K2-SA algorithm to learn the Bayesian Network structure. Experimental results show the effectiveness of the algorithm. In the later study and research, we will use this algorithm for applications, research more data mining classification methods and classification-oriented scoring method.

Acknowledgment. This work was supported in part by NSFC under Grant 61172090, Scientific and Technological Project in Shaanxi Province under Grant 2012K06-30.Thanks to the condition and environment provided by Shaanxi Key Lab of Computer Network. Thanks to the guidance of Pro. Xu and Pro. Gui.

References

1. Pearl, J.: Fusion, Propagation, and Structuring in Belief Networks. Artificial Intelligence 29, 241–288 (1986)
2. Wong, M.L., Leung, K.S.: An Efficient Data Mining Method for Learning Bayesian Networks using an evolutionary algorithm based hybrid approach. IEEE Transactions on Evolutionary Computation 8(4), 378–404 (2004)
3. Chen, X.W., Anantha, G., Lin, X.: Improving Bayesian Network Structure Learning With Mutual information-based node ordering in the K2 algorithm. IEEE Transactions on Knowledge and Data Engineering 20(1), 1–13 (2008)
4. Darwiche, A.: A Differential Approach to Inference in Bayesian Networks. arXiv preprint arXiv:1301.3847 (2013)
5. Lerner, B., Malka, R.: Investigation of the K2 Algorithm in Learning Bayesian Network classifiers. Applied Artificial Intelligence 25(1), 74–96 (2011)
6. Cooper, G., Herskovits, F.E.: A Bayesian Method for The Induction of Probabilistic networks from data. Machine Learning 9(4), 309–347 (1992)
7. Lam, W., Bacchus, F.: Learning Bayesian Belief Networks: all Approach Based on the MDL Principle. Computational Intelligence 10(4), 269–293 (1994)
8. De Campos, L.M., Juan, M.F., José, A.G., et al.: Ant Colony Optimization for Learning Bayesian networks. Computational Intelligence 10, 269–293 (1994)
9. Chickering, D.M.: Optimal structure identification with Greedy Search. The Journal of Machine Learning Researeh 3, 507–554 (2002)
10. Chickering, D., Geiger, D., Heckerman, D.: Learning Bayesian Networks: Search methods and experimental results. In: Proceedings of Fifth Conference on Artificial Intelligence and Statistics, pp. 112–128 (1995)

11. Aarts, E., Korst, J., Michiels, W.: Simulated annealing. Search methodologies, pp. 187–210. Springer, US (2005)
12. Steck, H.: Learning the Bayesian Network Structure: Dirichlet prior Versus Data. arXiv preprint arXiv:1206.3287 (2012)
13. Rissanen, J.: Minimum description length principle. Encyclopedia of Machine Learning, pp. 666–668. Springer, US (2010)
14. Lam, W., Bacchus, F.: Using Causal Information And Local Measures to Learn Bayesian networks. In: Proceedings of the Ninth International Conference on Uncertainty in Artificial Intelligence, pp. 243–250. Morgan Kaufmann Publishers Inc. (1993)
15. Yun, Z., Keong, K.: Improved MDL Score for Learning of Bayesian Networks. In: Proceedings of the International Conference on Artificial Intelligence in Science and Technology, AISAT, pp. 98–103 (2004)
16. Jiang, J., Wang, J., Yu, H., Xu, H.: Poison Identification Based on Bayesian Network: A Novel Improvement on K2 Algorithm via Markov Blanket. In: Tan, Y., Shi, Y., Mo, H. (eds.) ICSI 2013, Part II. LNCS, vol. 7929, pp. 173–182. Springer, Heidelberg (2013)
17. Lauritzen, S.L., Spiegelhalter, D.J.: Local Computations with Probabilities on Graphical structures and their application to expert systems. Journal of the Royal Statistical Society, Series B (Methodological), 157–224 (1988)
18. Beinlich, I.A., Suermondt, H.J., Chavez, R.M., Cooper, G.F.: The ALARM monitoring system: A Case Study with Two Probabilistic Inference Techniques for Belief Networks, pp. 247–256. Springer, Heidelberg (1989)