International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **11**, November **2013**

ICIC International ©2013 ISSN 1349-4198
pp. **4459–4469**

IMPROVED HYBRID METHOD FOR CONSTRUCTING SMALL AND MEDIUM-SIZED BAYESIAN NETWORKS

Liyan Dong[1,2], Yang Gao[1,2], Pengfei Yan[1,2], Zhen Li[1,2] and Yongli Li[3,*]

[1]College of Computer Science and Technology
[2]Symbol Computation and Knowledge Engineering of Ministry of Education
Jilin University
No. 2699, Qianjin Ave., Changchun 130012, P. R. China
dongliyan@gmail.com

[3]School of Computer Science and Information Technology
Northeast Normal University
No. 5268, Renmin Street, Changchun 130117, P. R. China
*Corresponding author: liyl603@nenu.edu.cn

Received September 2012; revised January 2013

Abstract. *This paper proposed a method for constructing small and medium-sized hybrid Bayesian networks (HBN) without any priori information. The method first adopted the idea of pseudo-BN. In Addition, we designed a novel scoring function and improved Silva's algorithm by finding an approximate minimum d-separating set blocking two designated nodes. The structure learning process showed that HBN model integrated the advantages of Naive Bayesian Classifier (NBC) model and Tree Augmented Bayes Network (TAN) model. The experimental result showed HBN model had good robustness and higher classification accuracy than both TAN and NBC at the expense of tolerable larger time cost. Lastly, a practical application of HBN algorithm further validated it for small and medium-sized networks.*
Keywords: Hybrid Bayesian networks, Structure learning, Pseudo-BN

1. **Introduction.** Bayesian networks integrate much knowledge from the probability theory and the graph theory. As shown in the work of [1-3], it consists of the qualitative component DAG (Directed Acyclic Graph) and the quantitative component CPT (Conditional Probability Table). DAG $= (V, A)$ is a directed acyclic graph, where the node set $V = \{x_i | 1 \leq i \leq n\}$ represents a variable set of the problem field to be tackled and each element of the arc set $A = \{u \to v | u, v \in V\}$ corresponds to the direct probabilistic dependence between two nodes. $CPT = \{f_{x_i} : \{x_i\} \times \prod_i \to [0, 1] | x_i \in V\}$ is a set of real-valued function, which is utilized to measure uncertainty of direct dependence between two nodes in V. Specially, each element of V corresponds to a CPT.

The structure learning of Bayesian networks aims to find a network structure, which possesses maximum posterior probability defined in the work of [4-6],

$$B_S^{MAP} = \arg\max_{B_S^h} \left(p(X|B_S^h) = \prod_{i=1}^{n} p\left(x_i | \prod_i\right)\right) \tag{1}$$

where $\prod_i \subseteq \{x_1, \ldots, x_{i-1}\}$ is a parent set of the node x_i and makes x_i conditionally independent of $\{x_1, \ldots, x_{i-1}\} \backslash \{\prod_i\}$.

We noted that the space of candidate network structures exponentially rises with the size of the node set V and states of each node. So the guiding ideology for constructing

Bayesian networks is that, to construct the most simple network structure that can decompose and express joint probability distribution of n variables effectively, namely, to compute a variable order (VO) according to Markov condition. However, it is a tickler for each node x_i to allocate the minimum node set $\prod_i \subset \Pr(x_i)$. Therefore, several methods have been usually used to tackle this problem as a tradeoff. For example, each node usually has no more than k parent nodes in most study.

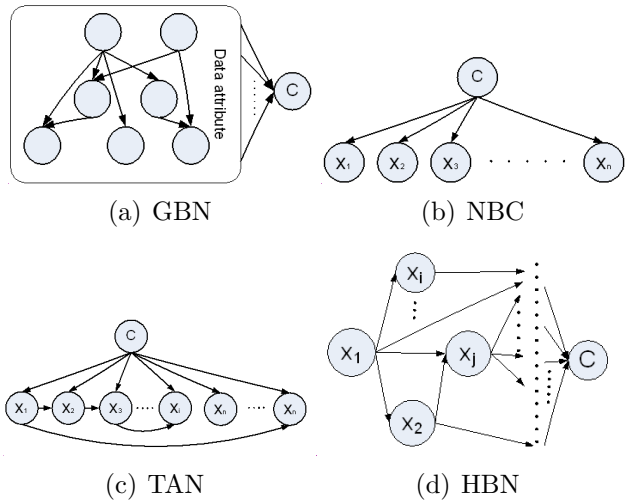
There are mainly three approximate solutions to the DAG learning problem. First, constraint-based methods utilize statistics or information theory to run conditional independence (CI) test and to analyze whether two variables are dependent. The reliability and efficiency of this solution mainly depend on the frequency and complexity of CI tests. Second, scoring-based methods aim to find an optimal network model G with the least arcs and best fit to data model D. The solution consists of a scoring metric and a search algorithm. The scoring metric, such as Bayesian method, MDL, entropy method and BDe scoring, is used to assess how well G fits to D. The search algorithm is applied to finding a network with maximum score. However, it is not realistic to perform an exhaustive and greedy search. Consequently, some inexact search methods have recently emerged, such as heuristic techniques, GA, simulated annealing and MCMC as shown in [7,8]. Lastly, hybrid approaches, attracting much interest of researchers, first utilize CI test to reduce the complexity of search space, and then performs the process of scoring and searching to find a suboptimal network structure using a unified ideology.

2. **Problem Statement and Preliminaries.** There have been many Bayesian network models with different characters which are as follows:

- whether direct probabilistic dependence existing among data nodes;
- which kind of structure of network model solely consisting of data nodes;
- whether classificatory node has brother nodes.

Especially for classification task, there are three prevalent Bayesian network models, such as Generalized Bayesian Networks (GBN), Naive Bayesian network Classifier (NBC) and Tree Augmented Naive Bayes (TAN) as shown in Figures 1(a)-1(c).

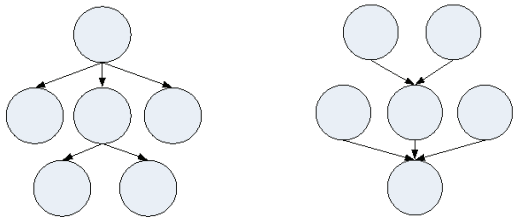
GBN model is a generalized Bayesian network in which there may be more than one data node with 0 in-degrees. As a dense DAG, GBN model is extremely difficult to construct. Besides, the computational complexity of classification is also beyond user's



(a) GBN (b) NBC

(c) TAN (d) HBN

FIGURE 1. Four different Bayesian network models



(a) Single-tree model (b) Poly-tree model

FIGURE 2. Two different models of data attributes

tolerance. In view of these factors, GBN model is actually solely regarded as a theoretical model. NBC model is a widely used classification model, as shown in the work of [9-11]. The model stipulates that all data attributes are independent of each other but each data node has direct probabilistic dependence on classificatory attribute. The structural simplicity makes construction of network easier. However, the hypothesis of mutually independent data nodes is always not established in reality, so the classification accuracy of NBC model is usually unsatisfactory. Given the disadvantages of NBC model, TAN model makes some improvement and allows direct probabilistic dependence among data nodes. As a trade-off between NBC and GBN, TAN model relaxes the limitation of the independence hypothesis, but regulates that each data node has one parent node at most. In the work of [12], TAN classifier was improved by integrating GA into Bayesian theory.

Our HBN model, as shown in Figure 1(d), is a re-expansion of TAN model. In addition to unique root node, it stretches the constraint and allows each node can have k parent nodes at most. Figure 2 gives two different models. Figure 2(a) is a single-tree model defined in the work of [13], which is a general tree model in which only one the root node exists, and each non-root node has one parent node at most. Figure 2(b) is a poly-tree model defined in the work of [14], which is simply a connected causal network in which each node may have many parent nodes. In the TAN, the model of data attributes is evidently a single-tree model. However, for HBN, the model of data attributes integrates characters of both single-tree model and poly-tree model.

3. **Methods.**

3.1. **Scoring function.** First, $Indent(u, v|CS)$ was defined as a conditional independence assert in data model D and its value could be attained by computing the conditional mutual information $I(u, v|CS)$ drawn from the work of [15], namely,

$$Indent(u, v|CS) = \begin{cases} 1, & \text{if } I(u, v|CS) = \text{true}; \\ 0, & \text{else.} \end{cases} \tag{2}$$

Thereafter, $\langle u, v|CS\rangle$ was defined as a conditional independence assert in network model, G and its value could be computed using classical d-separation criterion.

Based on $Indent(u, v|CS)$ and $\langle u, v|CS\rangle$, a function was defined to measure the extent of deference by expressing the order m conditional independence between the network model G and the data model D.

$$diff_m(G, M) = \sum_{x_i, x_j \wedge i \neq j} (Indent(x_i, x_j|CS) \oplus \langle x_i, x_j|CS\rangle) \tag{3}$$

Given intolerable computational complexity, the high order CI test is unreliable to some extent. So we only consider the situations of $m = 0$ and 1 in the above scoring function. Obviously, the more it fits to D G, the smaller the denominator of function $g(G)$ is, and the score of model G is larger.

3.2. **Finding minimum *d*-separating set.** In this section, an improved algorithm will be introduced to find an approximate minimum d-separating set S blocking two designate nodes u and v in DAG.

First, the following conclusion can be proved,

$$S = \{s | s \in ancestor(u, v)\} \tag{4}$$

where $ancestor(u, v) = ancestor(v) \cup ancestor(v)$. The establishment of this conclusion enormously reduces the computational complexity of finding minimum d-separating set.

Additionally, the idea of moral graph is introduced. The moral graph G^m of the directed graph G is an equivalent undirected version of G. As defined in the work of [16], given a directed acyclic graph $G = (V, E)$, its moral graph, $G^m = (V, E^m)$, is generated by connecting nodes with common children, and making all the edges in the undirected graph. Besides, in the work of [17], given $A(u, v) = \{u, v\} \cup ancestor(u, v)$, if $G^m_{A(u,v)}$ is the moral graph of sub-graph $G_{A(u,v)}$, then the process of finding minimum d-separating set blocking u and v in G is equivalent to that of finding minimum separating set blocking these two nodes in $G^m_{A(u,v)}$.

Ideally, we should find a minimum separating set $S_{\min}$ blocking u and v. However, since it is a combinatorial optimization problem to find the set $S_{\min}$, and the time complexity exponentially rises with node size, we have to find an approximate minimum separating set S as an acceptable trade-off.

At present, most of algorithms for finding set S are based on Silvia's algorithm in 1996, which is carried out through seeking all mutually disjoint shortest open paths between two nodes, namely,

$$ShortestOpenPathSet = \{p | \forall i, j \leq n \wedge i \neq j, p_i \cap p_j = \varnothing\} \tag{5}$$

since in the set, it is mutually disjoint between each two elements, which have shortest length. So these elements correspond to the elements of set S.

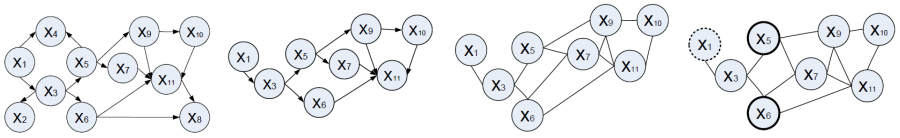
In the work of [17] (i.e., Silvia's algorithm), the process of finding each shortest open path between u and v contains two sub-processes: forward procedure and backward procedure. The forward procedure uses a labeling procedure to trace the disjoint path and adopts DFS strategy. From v, for each node w to be tested in some candidate path, we used the name of its some adjacent node as its negative label label-$n(w)$ until the current tested node is u or null; the backward search tries to recover the new path and the forward procedure ends at u. Besides, each node in this new path is marked with some found path, and direction of each edge is pointed out. The backward procedure may need to modify some found paths to meet the constraint of disjoint paths. The forward procedure and the backward procedure are executed repeatedly until all search trees are blocked before arriving node u.

The ideology of moral graph is also applied in our algorithm. The difference between our algorithm and Silvia's algorithm is the procedure of finding the set S blocking each two nodes in G. The following algorithm describes our procedure for finding the set S in more detail.

As shown in Figure 3, we demonstrate how to find the approximate minimum separating set S blocking x_3 and x_{11} in G on basis of our algorithm.

In detail, Figure 3(a) shows a directed graph G consisting of 11 nodes, and we try to seek the set S blocking x_3 and x_{11} in G. Figure 3(b), $G_{A(u,v)}$, is a sub-graph of Figure 3(a) and its node-set consists of all ancestor nodes of x_3 and x_{11}. Figure 3(c) is a moral graph of Figure 3(b). In Figure 3(d), the adjacent nodes $\{x_5, x_6\}$ of x_3 are chosen as the approximation of S, while x_1 is expelled from S in that it cannot reach x_{11}. As shown in Figure 3, with the improved algorithm, we obtain an approximate minimum d-separating

Algorithm MINIMUM_D-SEPARATION_SET
Input: graph $G = (E, V)$, u, v
1 Generate the set $A(u, v)$, namely, the ancestor nodes set of u and v;
2 Generate the sub-graph $G_{A(u,v)}$;
3 Generate the moral graph $G^m_{A(u,v)}$, which is equivalent to $G_{A(u,v)}$;
4 Choose the one with smaller degree from u and v in $G^m_{A(u,v)}$, and then initialize the set S with the adjacent nodes of the chose node;
5 From the set S, delete all nodes that cannot reach the other node, and the current generating set S is the one for the request.



(a) G (b) $G_{A(u,v)}$ (c) $G^m_{A(u,v)}$ (d) Result

FIGURE 3. A demonstration of finding the set S blocking x_3 and x_{11}

set $\{x_5, x_6\}$. The same result can be obtained using Silvia's algorithm. However, our algorithm is somewhat inferior to Silvia's algorithm on the accuracy of the result, because the cardinality of the set S we attained is slightly larger than Silvia's result. However, the complexity of our algorithm is distinctly superior to Silvia's algorithm and easier for implement. In addition, the result attained using our algorithm is acceptable, and so the algorithm could be viewed as a compromise solution.

3.3. **Algorithm HBN.** Algorithm HBN is a three-phase algorithm for constructing hybrid Bayesian networks.

In first phase, we generated VO, which is a sequence θ of data attributes. VO is considerably essential in the whole learning process, because it can effectively reduce search space, decide direction of arcs and guarantee acyclic character of network model. However, as pointed out by the work of [18], there may be many VOs corresponding to a specific potential network structure. And applying different VOs, a specific structural learning algorithm may generate different network structures. In fact, to obtain an optimum VO, it requires as much information as learning a complete structure, and the computation may be extremely complex as well. As a result, we had to use a suboptimum VO as an approximate optimum VO. Drawn from the work of [19,20], information gain (IG), originating from algorithm ID3, is adopted as an index for sorting data attributes in the study, where IG of data attribute x_i about category attribute C is defined as follows:

$$IG(C; x_i) = H(C) - H(C|x_i) = H(x_i) + H(C) - H(x_i, C) \tag{6}$$

where entropy $H(C)$ is the measure of the value of uncertainty of category attribute C, conditional entropy $H(C|x_i)$, is the measure of the value of uncertainty of C given the value of x_i.

The second phase aims to construct pseudo-BN in which the out-degrees of some nodes are 0 and no category attribute exists. The process can be implemented by three steps. First, the conditional mutual information was exploited to measure the probabilistic dependence between two nodes, and then algorithm MWST was used to generate frame of pseudo-BN. Second, we only added some necessary arcs into the network frame using a scoring-based method, thereby obtaining an I-map model of potential network model

$\tilde{G}$. Besides, an improved greedy search strategy was adopted. Before performing each search, each element $I(u, v|C)$ of the set was tested for deciding the candidate arc set $arcToAdd\_Set$:

$$arcToAdd\_Set = \{u \to v | I(u, v|C) \in S' \wedge I(u, v|C) > t_1 \wedge u \prec v \in \theta\} \tag{7}$$

When we solely considered adding arcs into network in the previous phase, the attained model G_2 may over-fit model M, and then deviate from model $\tilde{G}$. It is because the training data set D is merely the small section of mass data sampled by model $\tilde{G}$ and does not possess all characteristics of model $\tilde{G}$. According to the limit theory, the model M reflected by data set D is consistent with model $\tilde{G}$, if and only if the size of data set D is large enough. In fact, it is not realistic to use mass data for learning model G_2, so model G_2 generated using the training data D must deviate from model $\tilde{G}$. To overcome this drawback, we pruned G_2 through removing some weak dependence relations using a constraint-based method in the third step. Besides, the pruning was also beneficial to the computational complexity of classification.

In the last phase, the category attribute was brought into the ultimate network model in order to eliminate the fake of pseudo-BN. The operation of closing graph mainly takes two indexes into consideration, namely, mutual information $I(x_i, C)$ and OutDegree (x_i, C). The reason for considering OutDegree (x_i, C) was that the out-degree of data attribute should not be zero. Besides, we could control the arcs, link the category attribute C and some data attribute x_i, into the network model by considering $I(x_i, C)$.

The following algorithm HBN details the process of constructing a hybrid Bayesian network.

Algorithm HBN

Input: D, $X = \{x_1, \ldots, x_{n-1}, C\}$, where C is category attribute and x_i is a data attribute.

1 Generate variable order θ: compute the $IG(x_i)$ of each data attribute, and sort all data attributes according to their information gain.

2 Generate frame network G_1 with algorithm MWST.

3 Add the necessary arcs as following process:
   1) Initialize the set arcToAdd_Set;
   2) For each $e_i \in arcToAdd\_Set$, we first temporarily add it into graph; then compute the score of the graph and delete this arc from the graph; at last, we choose the arc, $u \to v$, which adding could make the graph possess maximum score. If the improvement is greater than some threshold, then we add this arc $u \to v$ to the graph, else, terminate the process of adding arc.
   3) If the in-degree is equivalent to k, then we delete all arcs which source is u from the arcToAdd_Set. Return 2).

4 Delete the redundant arcs using the following process:
   1) For each arc $u \to v$, which is added in precious step, we first find an approximate minimum d-separating set S, which could block u and v in graph, using algorithm Minimum_D-separation_Set.
   2) We use the set S as a condition set to conduct CI test. If the result of the test is true, we delete the arc $u \to v$ form the graph.
   3) The above process is executed repeatedly until new added arcs are all tested.

5 Close the graph, namely introduce the classification attribute C into graph: for each data attribute node x, we add the arc $x \to C$ into graph, if and only if the out-degree of node x is equivalent to 0, or the mutual information $I(x, C)$ is greater than some threshold.

---

4. **Experiments and Results.** To evaluate performance of our algorithm, experiments were conducted on two standard UCI datasets, Nurseryc and Chess as shown in Table 1.

TABLE 1. The experiment data

Data Set	Data Attribute	Classification Attribute	Data Size
Nurseryc	8	4	12960
Chess	9	2	3196

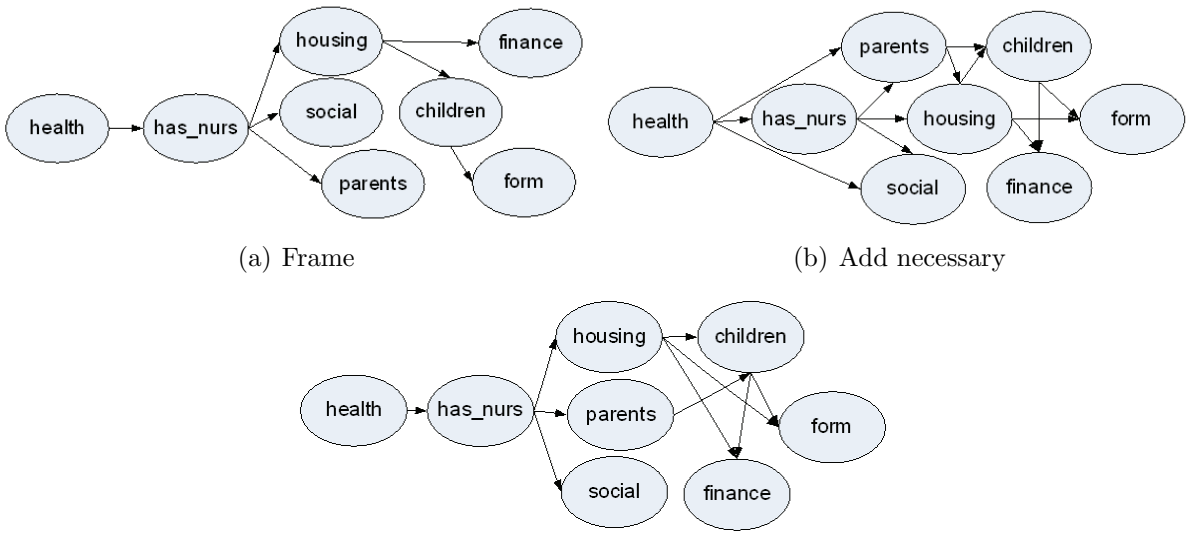
Especially, the data set Chess has much less records than the data set Nurseryc. And it is specially chosen for the purpose of validating robustness on classification accuracy of HBN model.

4.1. **Model constructing.** The Nurseryc problem was chosen to demonstrate the procedure of constructing an HBN model. And it involves 9 variables including 8 data attributes and one 4-value classification attribute. Using HBN algorithm, variable order can be first obtained as shown in Table 2.

TABLE 2. The variable order

Data Attribute	Information Gain (IG)	Variable Order (VO)
parents	0.074869	3
has_nurs	0.199002	2
form	0.005791	7
children	0.012187	6
housing	0.020069	5
finance	0.004303	8
social	0.022568	4
health	0.955067	1

After that, the pseudo-BN can be constructed by three steps as shown in Figure 4.



(a) Frame (b) Add necessary

(c) Delete redundant arc

FIGURE 4. Constructing pseudo-BN

For the construction of the pseudo-BN, in Figure 4(a), network frame is firstly generated using algorithm MWST; in Figure 4(b), six necessary arcs are then added into the frame by the search-and-score based method, including health → social, health → parents, parents → housing, parents → children, housing → form, and children → finance; in Figure 4(c), three redundant arcs are deleted from Figure 4(b) using the dependence analysis based method, including health → social, health → parents and parents → housing.

Thereafter, the classification attribute needs to be introduced into the network model G for the purpose of enclosing the pseudo-BN as demonstrated in Figure 5.

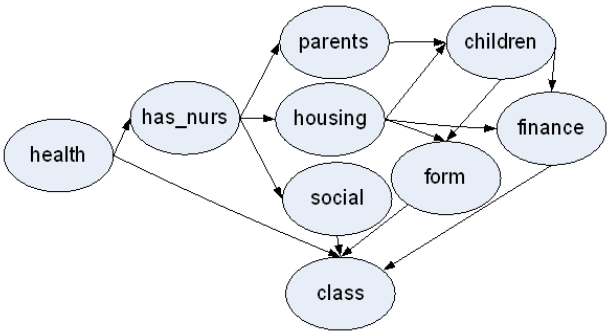


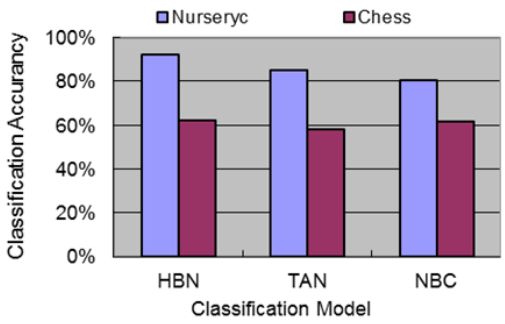
FIGURE 5. Close graph

4.2. **Model evaluating.** The HBN algorithm was evaluated from three aspects: classification accuracy, time cost and robustness.

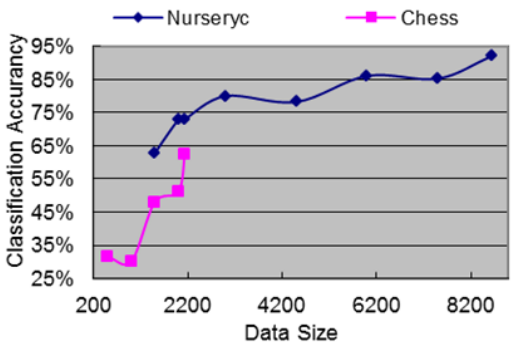
Classification accuracy was compared among three classification models, HBN, TAN and NBC. As can be seen from Figure 6(a), the model HBN produced higher accuracy than the others for the two data sets. In addition, for model HBN, the overall trend of the classification accuracy increases with the size of data set for both Nurseryc and Chess as shown in Figure 6(b).

As for robustness on classification accuracy of HBN model, the accuracy can be rapidly increased into a satisfactory level as shown in Figure 6(b). However, the classification accuracy of the HBN model trained on Chess is logically unsatisfactory due to the limited size.

For evaluation of time cost, the same models were used for comparison on the same data sets as above. As shown in Figure 7(a), HBN model has a much higher time cost than the others. And for large sized network, the effect of intolerable time cost on overall

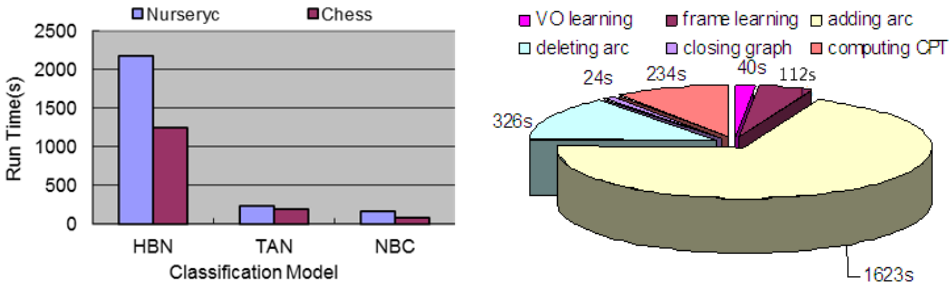


(a) Comparison of classification accuracy



(b) Robustness on classification accuracy of HBN model

FIGURE 6. Comparison and robustness of classification accuracy



(a) Comparison of run time

(b) Time-cost distribution of each constructing phase

FIGURE 7. Time cost of algorithm HBN

performance obliterates its promotion on classification accuracy to some extent. In more detail, Figure 7(b) indicates the time cost of each stage of learning process based on training data set Nurseryc_Train. The majority of time cost was consumed in the adding arcs phase. It was because a scoring function similar to climbing function and a greedy search strategy were adopt for the purpose of making graph model G fit to data model D as much as possible. The negative impact on the whole training time suggests that the algorithm HBN is more suitable for constructing a medium and small sized network model rather than a large sized one.

5. **Application.** To further validate HBN model for small and medium-sized Bayesian networks, we applied it into identification problem of major hazards in emergency response domain. The hazards in a system referred to some potential power or material, which can result in human injury, property damage and environmental pollution.

In more detail, a classification model was constructed to identify level of forest fire. As shown in Table 3, the problem involved five data attributes including Forest Area, Number of Resident, Wind Velocity, Humidity Level and Temperature Level, and one 3-value classification attribute Hazard Level.

TABLE 3. Forest fire problem

Forest Area (m^2)	Number of Resident /10^3	Wind Level (m/s)	Humidity Level (%)	Temperature Level (°C)	Hazard Level
			> 60		
$< 6.66 \times 10^6$	< 1	< 5.4	$50 \sim 60$	< 18	major
$6.66 \times 10^6 \sim 6.66 \times 10^7$	$1 \sim 100$	$5.4 \sim 10.7$	$40 \sim 50$	$18 \sim 25$	general
$> 6.66 \times 10^7$	> 100	$10.7 \sim 17.1$	$25 \sim 40$	> 25	none
		> 17.1	< 25		

An authorized database for forest fire identification was used to train and test a classification model in this study. We used 6234 records for training model and 4521 for test. The trained model is shown in Figure 8.

For the small and medium-sized network, total time for training was 1974s, which is completely acceptable especially for training a Bayesian classifier. Furthermore, the classification accuracy can reach 88.4% for the test data set.

6. **Conclusions.** In the study, an improved hybrid algorithm HBN was proposed to learn Bayesian belief networks. A scoring-based method was first used to find a good graphical representation G for data model D. Thereafter, a learning algorithm based on

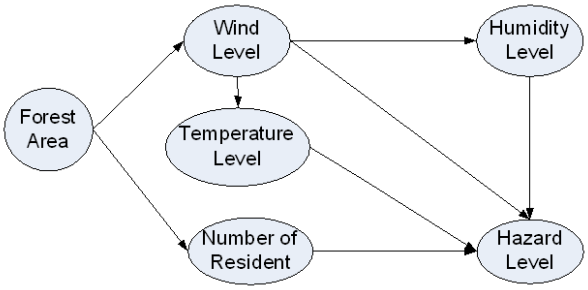


FIGURE 8. Classification model trained for the forest fire problem

independency relationships was utilized to prevent the graph model G from over-fitting data model D. The experiments demonstrated HBN model was superior to TAN and NBC on basis of classificatory accuracy. Additionally, in line with the comparison of time cost compared with the other two models, HBN algorithm is more suitable to construct small and medium-sized network. Besides, HBN model can possess some advantages of both inverse NBC and improved TAN by tuning thresholds involved in our algorithm. At last, the abstract problem of forest fire identification further validates HBN algorithm with acceptable time cost (1974s) and satisfactory classification accuracy (88.4%) for constructing small and medium-sized networks.

REFERENCES

[1] M. Pelikan, K. Sastry and D. E. Goldberg, iBOA: The incremental Bayesian optimization algorithm, *Proc. of Genetic and Evolutionary Computation Conference*, Atlanta, GA, USA, pp.12-16, 2008.

[2] S. Acid, L. M. de Campos and J. F. Huete, Estimating probability values from an incomplete dataset, *International Journal of Approximate Reasoning*, vol.27, no.2, pp.183-204, 2001.

[3] R. A. van Engelen, Approximating Bayesian belief networks by arc removal, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19, no.8, pp.916-920, 1997.

[4] D. Heckerman, *A Tutorial on Learning Bayesian Networks*, Microsoft Research, Microsoft Corporation, MSR-TR-95-06, 1995.

[5] G. F. Cooper and T. Dietterich, A Bayesian method for the induction of probabilistic networks form data, *Machine Learning*, pp.309-347, 1992.

[6] D. Heckerman, Bayesian networks for data mining, *Data Mining and Knowledge Discovery*, pp.79-119, 1997.

[7] X. Chen, G. Anantha and X. Lin, Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm, *IEEE Transactions on Knowledge and Data Engineering*, vol.20, no.5, pp.628-640, 2008.

[8] J. Zhang, S. Wang, L. Tian and N. Wang, Bayesian network learning for the medicine aid diagnosis analysis, *ICIC Express Letters, Part B: Applications*, vol.2, no.6, pp.1285-1290, 2011.

[9] A. Adam, M. I. Shapiai, Z. Ibrahim, M. Khalid and L. W. Jau, Development of a hybrid artificial neural network – Naive Bayes classifier for binary classification problem of imbalanced datasets, *ICIC Express Letters*, vol.5, no.9, pp.3171-3175, 2011.

[10] H. Tang, J. Wu, Z. Lin and M. Lu, An enhanced AdaBoost algorithm with naive Bayesian text categorization based on a novel re-weighting strategy, *International Journal of Innovative Computing, Information and Control*, vol.6, no.11, pp.5299-5310, 2010.

[11] L. Dong, S. Mo, X. Ren, Z. Li and H. Yin, Research on XML text categorization based on Bayesian classifier, *ICIC Express Letters*, vol.5, no.5, pp.1679-1684, 2011.

[12] L. Dong, H. Zhang, X. Ren and Y. Li, Classifier learning algorithm based on genetic algorithms, *International Journal of Innovative Computing, Information and Control*, vol.6, no.4, pp.1973-1981, 2010.

[13] J. Cerquides and R. L. Mantaras, TAN classifiers based on decomposable distributions, *Machine Learning*, vol.59, no.3, pp.323-354, 2005.

[14] J. M. Pena, Reading dependencies from Polytree-like Bayesian networks, *Proc. of the 23th Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon, pp.303-309, 2007.

[15] J. Cheng, R. Greiner, J. Kelly, D. Bell and W. Liu, Learning Bayesian networks from data: An information-theory based approach, *Artificial Intelligence*, vol.137, no.1-2, pp.43-90, 2002.

[16] T. Romero and P. Larraoaga, Triangulation of Bayesian networks with recursive estimation of distribution algorithms, *INT. J. APPROX. REASON*, vol.50, no.3, pp.472-484, 2009.

[17] S. Acid and L. M. de Campos, An algorithm for finding minimum d-separating sets in belief networks, *Proc. of the 12th Conference on Uncertainty in Artificial Intelligence*, Portland, Oregon, pp.3-10, 1996.

[18] S. Acid, L. M. de Campos and J. F. Huete, The search of causal orderings: A short cut for learning belief networks, *Proc. of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Toulouse, France, pp.216-227, 2001.

[19] X. Shao, G. Zhang, P. Li and Y. Chen, Application of ID3 algorithm in knowledge acquisition for tolerance design, *Journal of Materials Processing Technology*, vol.117, no.1-2, pp.66-74, 2001.

[20] E. R. Hruschka Jr. and N. F. F. Ebecken, Towards efficient variables ordering for Bayesian networks classifier, *Data & Knowledge Engineering*, vol.63, no.2, pp.258-269, 2007.