

Research Article

Bayesian Network Structure Learning by Ensemble Learning and Frequent Item Mining

Guoxin Cao¹ and Haomin Zhang ^{1,2}

¹College of Science, Guilin University of Technology, Guilin 541000, China

²Guangxi Colleges and Universities Key Laboratory of Applied Statistics, Guilin University of Technology, Guilin 541000, China

Correspondence should be addressed to Haomin Zhang; zhanghm@glut.edu.cn

Received 10 September 2022; Revised 27 October 2022; Accepted 24 November 2022; Published 11 February 2023

Academic Editor: Abolfazl Gharaei

Copyright © 2023 Guoxin Cao and Haomin Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the common problem of low learning effect in single structure learning of a Bayesian network, a new algorithm EF-BNSL integrating ensemble learning and frequent item mining is proposed. Firstly, the sample set is obtained by sampling the original dataset using Bootstrap, which is mined using the Apriori algorithm to derive the maximum frequent items and association rules so that the black and white list can be determined. Secondly, considering that there may be wrong edges in the black and white list, the black and white list is used as the penalty term of the BDeu score and the initial network is obtained from the hill climbing algorithm. Finally, repeat the above steps 10 times to obtain 10 initial networks. The 10 initial networks were integrated and learned by the integrated strategy function to obtain the final Bayesian network. Experiments were carried out on six standard networks to calculate F_1 score and HD. The results show that the EF-BNSL algorithm can effectively improve F_1 score, reduce HD, and learn the network structure that is closer to the real network.

1. Introduction

Bayesian network (BN), a probabilistic graphical model, was proposed by Pearl [1] in 1988 and is capable of effective inference and analysis of uncertain knowledge, which is one of the hot spots of research in machine learning. BN has a solid theory base and has been broadly applied in many industries, including transportation [2], industrial production [3–5], economy [6–8], medicine [9], and agriculture [10, 11]. It is obvious that the innovative research on BN theories and methodologies, as well as the scientific construction of more effective algorithms, will definitely promote the ability of problem solving in practical areas [12, 13].

The fundamental theory of BN mainly contains structure learning, parameter learning, and Bayesian inference. Parameter learning is to learn network parameters on the known structure of the network. Structure learning includes the learning of parameters and the learning of the network structure, which is the focus and difficulty of BN learning. The basis of parametric learning and Bayesian inference is

BN structure learning. However, finding the optimal structure of the BN is a NP-hard problem [14]. The computation complexity grows exponentially as the number of nodes increases.

There are three main structure learning methods, which are constraint-based, fraction-based, and hybrid methods. Constraint-based methods usually use conditional independence tests or mutual information to identify dependency relationships between variables. Spirtes and Meek [15] proposed the SGS algorithm, the first structure learning algorithm, which determines the network structure mainly the conditional independence between nodes, but the learning efficiency grows exponentially. Spirtes et al. [16] improved the SGS algorithm by proposing the PC algorithm, which is enabled to construct BN from sparse networks and allows the use of chi-square tests without the necessity of a specific independence test. Qi et al. [17] proposed the WMIF algorithm to learn the network structure using the weakest-first strategy. The core concept of the score-based approach is to find the best structure based on the scoring function by

traversing all possible structures. Cooper and Herskovits [18] proposed the K2 algorithm, but the algorithm required a prior upper limit on the ordering of the nodes and the number of parents per node. Lee and Beek [19] proposed a local greedy search method coupled with a perturbation factor and used the idea of a knockdown algorithm to improve the performance of a local greedy search by using a metaheuristic. The scoring functions commonly used are MDL [20], AIC [21], BIC [22], and BDeu [23]. The common search algorithms are the K2 algorithm [18], hill climbing algorithm [24], and genetic algorithm [25]. The essential idea of hybrid learning is to decrease the size of the search space by first testing for independence and then using a scoring search method to gain the most optimal network structure. The first hybrid learning algorithm was the CB algorithm proposed by Singh and Valtorta [26], which starts by attempting a constraint-based PC algorithm to identify the order of the nodes and then learns the structure using the K2 algorithm. Alonso-Barba et al. [27] proposed the I-ACO-B algorithm, which first reduces the complexity of the search space using independence tests, and then uses an ant colony algorithm for scoring search to obtain the optimal network structure.

Recently, Eggeling et al. [28] proposed the idea of introducing structural priors and Wang et al. [29] proposed the use of constraints, both of which can greatly decrease the search space of structure learning, which is an idea for improving the existing structure learning algorithms. Xiao et al. [30] proposed an algorithm to merge association rules and knowledge for network structure learning. Sun et al. [31] proposed a new hybrid approach by integrating the PC and PSO algorithm, which takes parts of the output of the PC algorithm as structure priors to improve the initial solutions to BN structure learning. Li et al. [32] proposed BN structure learning based on frequent term mining. Wang and Qin [33] proposed BN structure learning based on ensemble and feedback strategies, but only integrated multiple BNs are

obtained through BDeu score. Based on this paper, a new BN structure learning algorithm named the EF-BNSL algorithm based on ensemble learning and frequent item mining is designed. The EF-BNSL algorithm flowchart is shown in Figure 1.

2. Materials and Methods

2.1. Bayesian Network. A BN can be shown by $BN = (G, P)$, where $G = (V, E)$ means the directed acyclic graph (DAG), $V = \{x_1, x_2, \dots, x_n\}$ represents the set of nodes, E represents the set of directed edges, and P is the probability distribution among the nodes, which is commonly represented by the conditional probability table (CPT).

The joint probability distribution of the node set $V = \{x_1, x_2, \dots, x_n\}$ can be represented as

$$P(x_1, x_2, \dots, x_n) = \prod P(x_i | pa(x_i)), \quad (1)$$

where $pa(x_i)$ is the parent node of x_i and $P(x_i | pa(x_i))$ is a conditional probability. For node x_i , we consider the relationship with $pa(x_i)$, and the number of parameters is much smaller than computing the joint probability directly.

As shown in Figure 2, the nodes A and B have no parent node, so their probabilities are directly $P(A)$ and $P(B)$. The parents of node C are nodes A and B , so the probability of node C is $P(C|A, B)$. Similarly, the parent node of the node D is the node C . The probability of node D is $P(D|C)$. Finally, we can get the CPTs of all nodes.

The goal of Bayesian network structure learning is to obtain the most suitable network structure for the dataset. The scoring function is used to measure the fitness of the BN structure for the dataset. The search algorithm is a search strategy and can find the best BN structure when the scoring function is determined. In this paper, we chose the BDeu score [23]. The BDeu score is given as follows:

$$F(G | D) = \log (P(G)) + \sum_{i=1}^n \sum_{j=1}^{q_i} \left[\log \left(\frac{\Gamma(\eta_{ij})}{\Gamma(\eta_{ij} + N_{ij})} \right) + \sum_{k=1}^{r_i} \log \left(\frac{\Gamma(\eta_{ijk} + N_{ijk})}{\Gamma(\eta_{ijk})} \right) \right], \quad (2)$$

where $P(G)$ is the prior probability, n is the number of nodes, q_i is the number of parents of the i node, r_i is the number of values taken by the i node, N_{ijk} is the number of samples, which satisfy the node $x_i = k$, $pa(x_i) = j$ in the dataset, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, $\eta_{ij} = \sum_{k=1}^{r_i} \eta_{ijk}$. Usually, we assume that $\eta_{ijk} = \eta / r_i q_i$, where η is the given equivalent sample size.

After choosing the scoring function, the problem of structure learning is transformed into an optimization problem which we must find the structure with the highest

score among the possible structures. In this paper, we chose the hill climbing search algorithm [24].

2.2. Analysis of Association Rules. Association rules can be applied to identify association relationships between variables. The rule form of the association rule is $(V_i \longrightarrow V_j)$, which indicates the association rule of the i variable with the j variable, and V_i is called the premise of the association rule, and V_j is called the result of the association rule. The

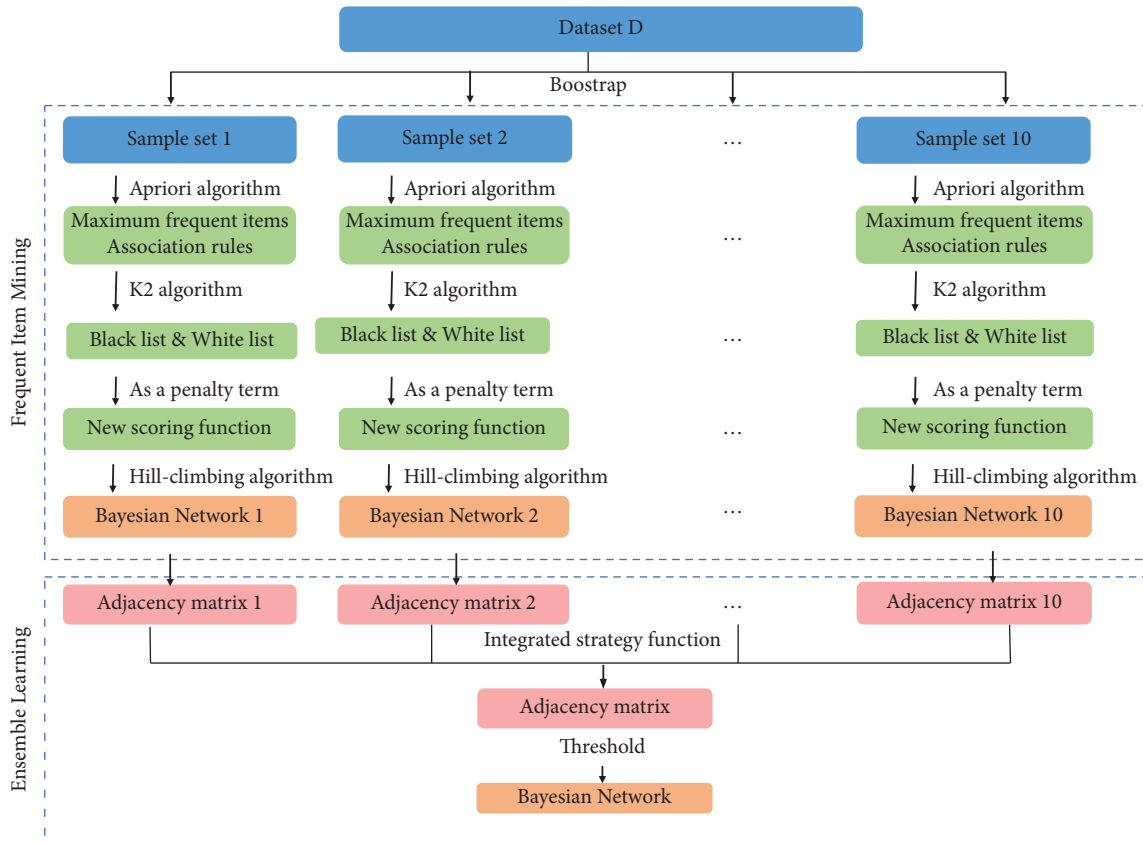


FIGURE 1: The flowchart of the EF-BNSL algorithm.

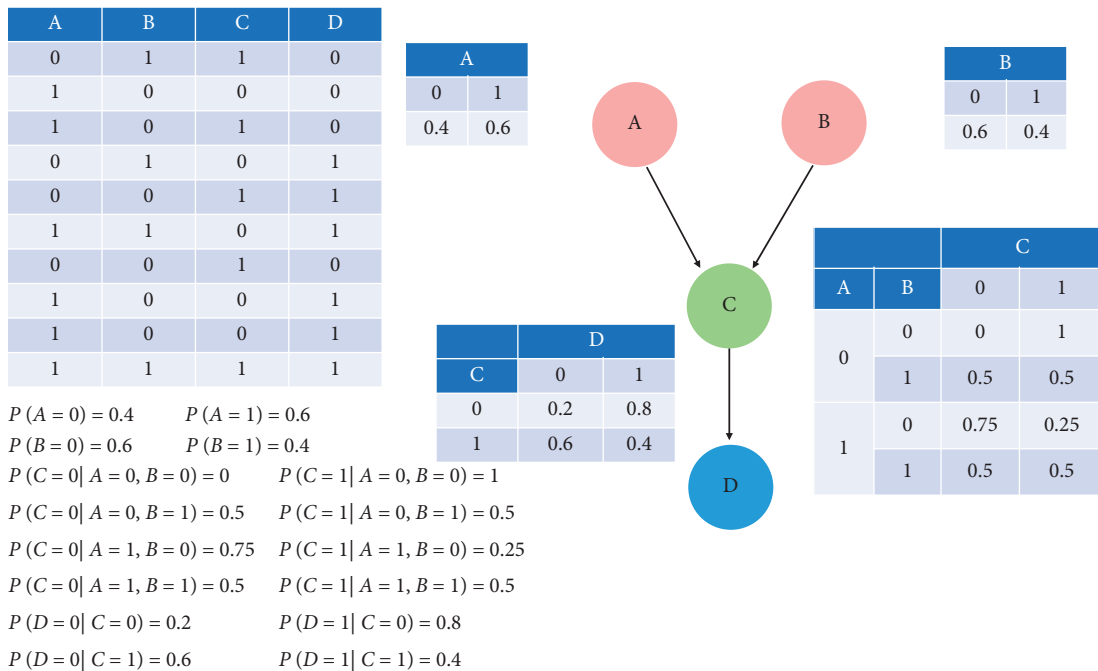


FIGURE 2: An illustrative example of the BN structure and its associated CPT.

definition of association rules requires the introduction of support and confidence, which is defined as follows:

$$\text{Support}(V_i \longrightarrow V_j) = \text{Support}(V_i \cup V_j) = P(V_i V_j), \quad (3)$$

$$\text{Confidence}(V_i \longrightarrow V_j) = \frac{\text{Support}(V_i \longrightarrow V_j)}{\text{Support}(V_i)} = P(V_j | V_i), \quad (4)$$

where $\text{Support}(V_i \longrightarrow V_j)$ is the support of the association rule $(V_i \longrightarrow V_j)$, which is the probability $P(V_i V_j)$ of V_i and V_j occurring simultaneously. $\text{Confidence}(V_i \longrightarrow V_j)$ is the confidence of the association rule $(V_i \longrightarrow V_j)$, which is the conditional probability of V_j occurring when V_i occurs.

The basic idea of the Apriori algorithm is to calculate the support of the item set by scanning the dataset several times and finding all frequent item sets to generate association rules. The set1 is said to be a superset of the set2 if every element of the set2 is in the set1. If all supersets of the frequent item set are nonfrequent item sets, the frequent item set is a maximal frequent item (MFI) set. The Apriori algorithm can mine the frequent item sets with support greater than the minimum support and find the MFI by filtering the association rules that satisfy $\{(V_i \longrightarrow V_j) | \text{Support}(V_i \longrightarrow V_j) \geq \text{min_support} \cap \text{Confidence}(V_i \longrightarrow V_j) \geq \text{min_confidence}\}$ to obtain the strongly associated rule set $\text{Ass} = \cup (V_i \longrightarrow V_j)$.

2.3. Ensemble Learning. Ensemble learning is a new machine learning method which is widely used for classification and regression tasks. Ensemble learning means using multiple identical or different learning algorithms to solve a unified problem by some combination [33].

The most classic ensemble learning is boosting and bagging. The boosting algorithm generates a training set and trains the model by sampling the original dataset with put-back, while the bagging algorithm generates multiple datasets by sampling them with a repeatable sampling technique (bootstrap) and trains the model on them separately and then combines the multiple models to obtain a more stable model. The primary aim of BN structure learning is to determine the directed edges between nodes, and bagging can effectively reduce the possible multilateral and antiedge problems, making the learning results more stable and reliable [34, 35].

3. EF-BNSL Algorithm

3.1. Building the Initial Network. Since the K2 algorithm is more sensitive, a certain incorrect result may mislead the construction of the whole BN, so this study uses the strong association rule set Ass to correct the BN structure of the MFI set Max_freq_item to increase its robustness.

The process is as follows: firstly, the variables in the MFI set are selected sequentially from the data $D = \{V, D_v\}$ and $D_{\text{Max_freq}} = \{D = \{V, D_v\} | V \in \text{Max_freq_item}_i\}$. Secondly, the BN structure $\text{BN}_{\text{freq_}i}$ is obtained by using the K2 algorithm for $D_{\text{Max_freq}}$, respectively, $\text{BN}_{\text{max_}f} = \cup \text{BN}_{\text{freq_}i}$. Lastly, if the node pair $(V_i \longrightarrow V_j) \in \text{BN}_{\text{max_}f}$ and $(V_i \longrightarrow V_j) \in \text{Ass}$, then the node pair is added to the white_list. Otherwise, it is considered impossible to have a dependency and is added to the black_list.

The pseudocode of the black and white list algorithm is listed in Algorithm 1.

Since the learned edges in the black and white list may have errors, a penalty term can be set in order to give the model certain error tolerance. A new score function is obtained by incorporating the black and white list as a penalty term. Finally, the initial BN is obtained from the hill climbing search algorithm. The new score function $F_{\text{new}}(G|D)$ is given as follows:

$$F_{\text{new}}(G|D) = F(G|D) + \omega \Phi((G|D)) = F(G|D) + \omega \left(\sum_{k=1}^{r_w} n_k - \sum_{t=1}^{r_b} n_t \right), \quad (5)$$

where $F(G|D)$ is the BDeu score, ω is the given weight, $\Phi((G|D))$ is the penalty term, r_w is the number of elements of the node pair $(V_i \longrightarrow V_j) \in \text{white_list} \cap (V_i \longrightarrow V_j) \in G_{PC}$, r_b is the number of elements of the node pair $(V_i \longrightarrow V_j) \in \text{black_list} \cap (V_i \longrightarrow V_j) \in G_{PC}$, n_k is the number of samples of the node pair $(V_i \longrightarrow V_j) \in \text{white_list} \cap (V_i \longrightarrow V_j) \in G_{PC}$, and n_t is the number of samples of the node pair $(V_i \longrightarrow V_j) \in \text{black_list} \cap (V_i \longrightarrow V_j) \in G_{PC}$.

3.2. Ensemble Learning. In this paper, we adopt the idea of ensemble learning and use an integrated strategy function W for calculating the final score of each edge. The integrated strategy function W is defined as follows:

$$W = \sum_{i=1}^n \sqrt{\frac{N_{\text{node}}}{N_{ki}}} M_{ki} \odot \frac{\sum_{i=1}^n M_{ki}}{n} \log(B), \quad (6)$$

where n is the number of samples, N_{ki} is the number of edges of the BN $G_{\text{max_}i}$ by combining the initial network using the i training sample, N_{node} is the number of nodes in the dataset, M_{ki} is the adjacency matrix of the BN $G_{\text{max_}i}$

```

(i) Input: Dataset  $D = \{V, D_v\}$ , MFI set Max_freq_item
      strong associated rule set Ass
(ii) Output: white list white_list, black list black_list
(1)   For  $i$  in range(len(Max_freq_item)):
(2)     Data_node = data[Max_freq_item[i]]
(3)     BN_freq_i = K2(Data_node)//BN obtained using K2 algorithm
(4)     For edge in BN_freq_i.edges(): //Loop BN of edges
(5)       If edge in Ass: //Determine whether the edges of a BN are in Ass
(6)         white_list ← edge//Get a white list
(7)     BN_max_f =  $\cup$  BN_freq_i//Aggregate the set of BN structures
(8)     If  $(V_i \rightarrow V_j) \notin \text{BN}_{\max\_f} \cap (V_i \rightarrow V_j) \notin \text{Ass}$ :
(9)       black_list ←  $(V_i \rightarrow V_j)$ //Get black list

```

ALGORITHM 1: Black and white list algorithm.

```

Input: Dataset  $D$ , the scoring function  $F_{\text{new}}(G|D)$ , threshold  $\theta$ 
Output: Bayesian network BN
(1)   For  $i$  in range (10): //Loop 10
(2)      $D_i \leftarrow \text{Bootstrap}(D)$ //Bootstrap sampling
(3)      $G_{\max\_i} \leftarrow \text{HillClimbSearch}(D_i, F_{\text{new}}(G|D_i))$ //Obtain the initial BN
(4)      $M_{ki} \leftarrow G_{\max\_i}$ //Represented by adjacency matrices
(5)      $M_s = M_s + M_{ki}$ //Count the number of times each edge has been learned
(6)      $M_x = \text{sqrt}(N_{\text{node}}/N_{ki} * M_{ki})$ //Assign weights to each edge
(7)      $M_k = M_k + M_x$ //Calculate the total weight value of each edge
(8)      $W_k = M_k \odot (M_s)/(n \log(D))$ //Calculate the integrated strategy function  $W_k$ 
(9)      $W_k = \text{normalize}(W_k)$ //normalized by the maximum-minimum
(10)  For  $p$  in range ( $\text{row}(W_k)$ ):
(11)    For  $q$  in range ( $\text{ncol}(W_k)$ ):
(12)      If  $W_k[p, q] < \theta$ ://Control the number of edges by the threshold  $\theta$ 
(13)         $W_k[p, q] = 0$ //0 means that no edge exists
(14)      Else:
(15)         $W_k[p, q] = 1$ //1 denotes the existence of directed edges ( $p \rightarrow q$ )
(16)  BN ←  $W_k$ //The adjacency matrix is transformed into a BN

```

ALGORITHM 2: EF-BNSL algorithm.

obtained using the i training sample, B is the sample size, and \odot is the Hadamard product.

The EF-BNSL algorithm flow is as follows: firstly, the original dataset is sampled 10 times using bootstrap, and each of the 10 sample sets is learned to obtain 10 initial BNs. The 10 initial BNs are denoted by the adjacency matrices as M_{ki} .

Secondly, the 10 adjacency matrices M_{ki} are integrated by the integration strategy function to obtain the score matrices W_k for each edge. The score matrix W_k is normalized by the maximum-minimum.

Thirdly, set the threshold θ , if the condition $W_k[p, q] < \theta$ is satisfied then we can set $W_k[p, q] = 0$, if the condition $W_k[p, q] > \theta$ is satisfied then $W_k[p, q] = 1$. We can get the adjacency matrix W_k .

Lastly, iterate over the adjacency matrix W_k , if $W_k[p, q] = 1$, then add the directed edge ($p \rightarrow q$), and finally get the ensemble learning BN. The number of edges of the BN can be effectively controlled by setting the size of the threshold θ . If the threshold θ is set too small, there will be many of redundant edges, and if it is set too large, the situation of fewer edges will be very dreadful.

The pseudocode of the EF-BNSL algorithm is listed in Algorithm 2.

4. Experiments and Result Analysis

4.1. Experiment Preparation. In this paper, we use the python integrated environment Anaconda3 with python version 3.8.5 for programming. Six standard networks of different sizes were downloaded from the Bayesian Network Repository [36]. BN was loaded through the pyAgrum package and the corresponding size dataset was generated, frequent term mining and association rule analysis were performed through the mlxtend package, the sklearn package for Bootstrap sampling, and the pgmpy package for the BN structure learning.

Three networks of different types were chosen for the experiment, including the small size networks (up to 20 nodes), Asia and Sachs, the medium size networks (20–50 nodes), Alarm and Insurance, and the large size networks (50–100 nodes), and Hailfinder and Hepar2. The six standard networks are shown in Table 1.

TABLE 1: The six standard networks used in the experiments.

Standard network	Type	Sample size	Nodes	Edges
Asia	Small size	20000	8	8
Sachs	Small size	20000	11	17
Alarm	Medium size	20000	37	46
Insurance	Medium size	20000	27	52
Hailfinder	Large size	20000	56	66
Hepar2	Large size	20000	70	123

4.2. *Evaluation Indicators.* To verify the effectiveness of the EF-BNSL algorithm, the F_1 score and the Hamming distance (HD) are chosen to evaluate the generated BN and defined as follows:

$$\text{recall} = \frac{TP}{(TP + FN)}, \quad (7)$$

$$\text{precision} = \frac{TP}{(TP + FP)}, \quad (8)$$

$$F_1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}, \quad (9)$$

$$HD = FP + FN, \quad (10)$$

where TP is the number of edges in both the current network and the standard network, FP is the number of edges in the current network but not in the standard network, and FN is the number of edges that do not exist in the current network but exists in the standard network, recall is the recall rate, and precision is the precision rate.

The F_1 score is a combination of recall rate and precision rate. A larger F_1 score means that the BN is closer to the real network. The HD is the sum of FP and FN. A smaller HD means a smaller number of erroneous edges and indicates that the learned BN is closer to the standard network.

4.3. *Results.* To verify the performance of the EF-BNSL algorithm, we conducted experiments on six networks. The following is an example of the Asia network. The Asia network is a small BN, which is used as a fictional medical example. The network asks whether a patient has tuberculosis, lung cancer, or bronchitis and consists of eight nodes and eight edges. Each random variable is discrete and can take two states. The original Asia standard network is shown in Figure 3.

The experimental process is divided into three stages. The first stage of the experiment is to perform frequent item mining and association rule analysis on the sample set. Firstly, we load the BIF file to generate a dataset with a sample size of 20000, and then we perform sampling by using bootstrap with a sample size of 1000. The sample set variables are transformed into Boolean variables. The minimum support and minimum confidence were set to 0.75 and 0.95 [32], respectively. The minimum support and minimum confidence can control the number of edges in the black and white list, which can be selected on the basis of the dataset distribution characteristics.

By using the Apriori algorithm with the given minimum support, the MFIs and association rules can be obtained. An association rule with a confidence level greater than the given minimum confidence is a strong association rule.

The MFI set Max_freq_item is {lung, tub, asia, xray, either}, and the strong association rule results are shown in Table 2.

On the basis of obtaining the MFI set and strong association rule results, the black and white list can be obtained from the proposed Algorithm 1 above. The black and white list of results are shown in Table 3.

In this experiment, the edges (either, xray) and (lung, either) in the white list are in the original Asia standard network, and the accuracy of the white list reaches 100%. On the other hand, all the edges in the black list are judged correctly except for the edge (tub, either) error, and the accuracy rate of the black list reaches 85.7%. To give the model some fault tolerance, so we do not use the list directly as the black and white list of BN. Instead, they are given certain weight scores and added to the scoring function as penalty terms. Also, we find that the number of edges in the white list is smaller than the number of edges in the black list in the experiment. Different weights can be further set for the edges of the black and white list in future studies.

The second stage of the experiment is to update the scoring function by the black and white list and get the initial network. Firstly, according to (3), the black and white list is added to the scoring function as a penalty term. The penalty term weight ω was set to 0.5 [32]. If an edge in the black list is contained in BN, the new scoring function becomes smaller. Conversely, if the edges in the white list are contained in BN, the new scoring function becomes larger. The network of the maximum score is obtained from the hill climbing search algorithm, which is the initial BN.

As shown in Figure 4(a), the BN is a network obtained by direct structure learning using BDeu score, which has 14 edges. Figure 4(b) shows the CPT of this BN structure. For example, the lung node has only tub node as its parent node, so there are four possible outcomes for the CPT of the lung node. $P(\text{lung} = 0 | \text{tub} = 0) = 0$, $P(\text{lung} = 1 | \text{tub} = 0) = 1$, $P(\text{lung} = 0 | \text{tub} = 1) = 0.0633$, and $P(\text{lung} = 1 | \text{tub} = 1) = 0.9367$. As shown in Figure 5(a), the BN is a network obtained by structure learning using an updated scoring function, with has 12 edges. Figure 5(b) shows the CPT of this BN structure.

By comparing with the original Asia standard network, we can able to calculate the F_1 score and HD for these two BNs using equations (7)–(10). The F_1 score and HD of the BN in Figure 4(a) is 0.45 and 9, respectively. The F_1 score

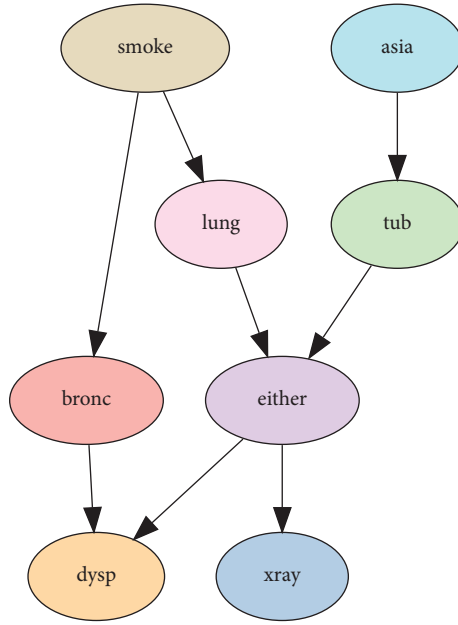


FIGURE 3: The original Asia standard network.

TABLE 2: The strong association rule results of the Asia network.

Association rule	Support	Confidence
(Either, xray)	0.8880	0.9579
(tub, Asia)	0.9760	0.9869
(Xray, Asia)	0.8800	0.9888
(Asia, tub)	0.9760	0.9889
(Either, Asia)	0.9170	0.9892
(Lung, Asia)	0.9260	0.9893
(Lung, either)	0.9270	0.9904
(Lung, tub)	0.9270	0.9904
(Xray, either)	0.8880	0.9978
(Xray, lung)	0.8880	0.9978
(Either, lung)	0.9270	1.0000
(Either, tub)	0.9270	1.0000
(Xray, tub)	0.8900	1.0000

TABLE 3: Asia network’s black list and white list.

Black list	White list
(Lung, xray)	(Either, xray)
(Asia, xray)	(Lung, either)
(Asia, lung)	
(Tub, lung)	
(Tub, either)	
(Asia, either)	
(Tub, xray)	

and HD of the BN in Figure 5(a) is 0.5 and 7, respectively. Improved scoring functions for BN learning outperformed the BN obtained by direct learning using the BDeu score.

The third stage of the experiment is to use Algorithm 2 to obtain the final ensemble learning BN. Firstly, bootstrap was used to sample the dataset for 10 times, and the 10 sample sets were, respectively, learned to obtain 10 initial BNs.

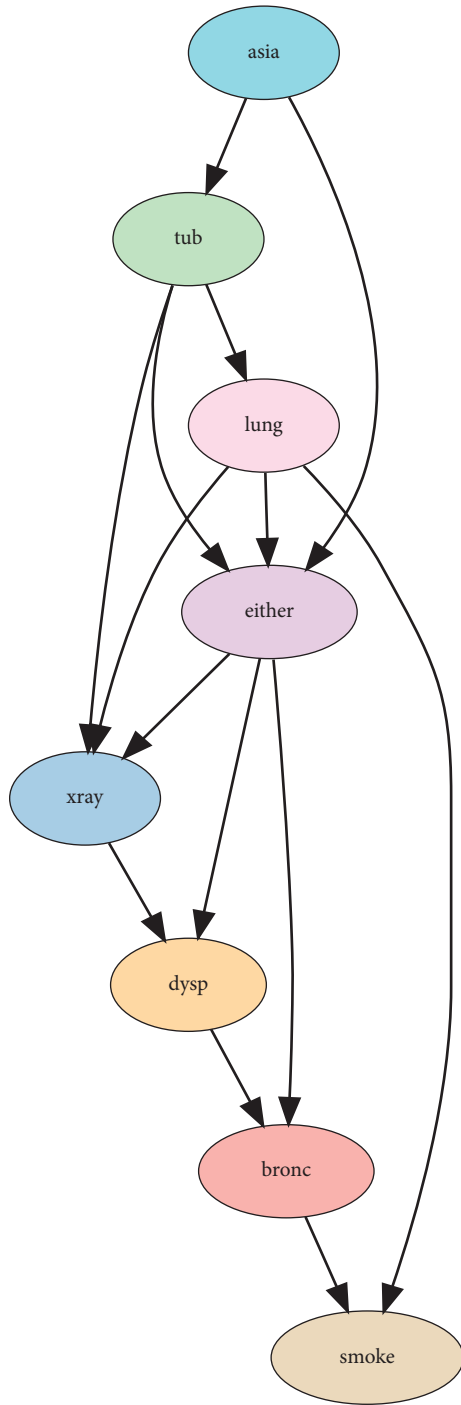
Secondly, the 10 initial BNs were denoted by the adjacency matrices as M_{ki} , and then the 10 adjacency matrices

M_{ki} were integrated by the integration strategy function to obtain the score matrix W_k . Meanwhile, the score matrix W_k was normalized by the maximum-minimum.

Thirdly, set the threshold $\theta = 0.4$ [33], if the condition $W_k[p, q] < 0.4$ was satisfied, then we can set $W_k[p, q] = 0$; if the condition $W_k[p, q] > 0.4$ was satisfied then $W_k[p, q] = 1$. We can get the adjacency matrix W_k .

Lastly, iterate over the adjacency matrix W_k . If the condition $W_k[p, q] = 1$, we can add the directed edge ($p \rightarrow q$) in the network. The final ensemble learning BN was obtained.

The final ensemble learning BN is shown in Figure 6(a), which has 10 edges. Figure 6(b) shows the CPT of this BN structure. For example, the xray node has only either node as its parent node, so there are four possible outcomes for the CPT of the xray node. $P(xray = 0either = 0) = 0.9706$, $P(xray = 1either = 0) = 0.0294$, $P(xray = 0either = 1) = 0.0483$, and $P(xray = 1either = 1) = 0.9517$. If we know that the value of node either is 0, then we can deduce that the probability that node Xray is equal to 0 is 0.9706. The F_1



(a)

FIGURE 4: Continued.

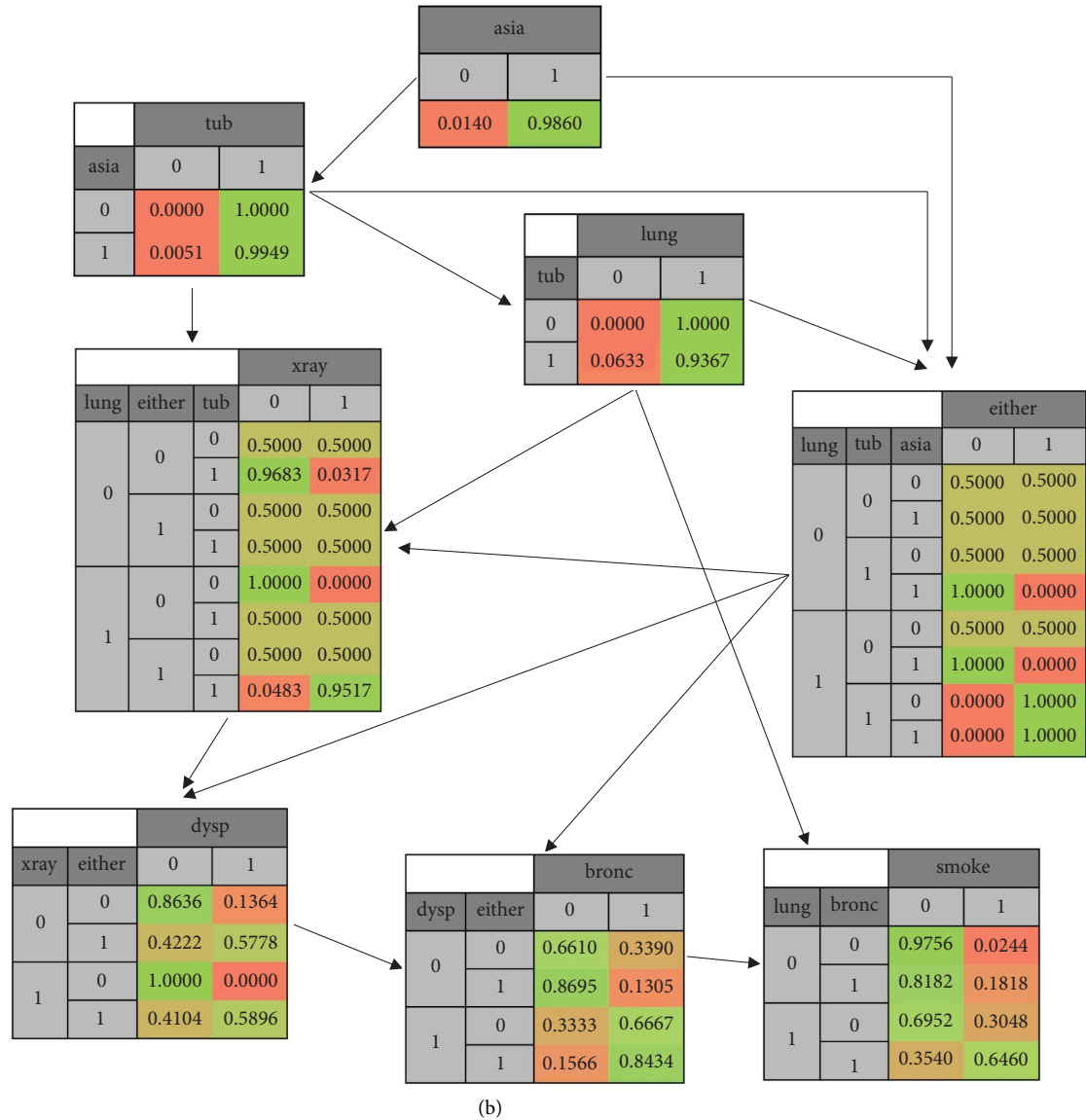


FIGURE 4: (a) The BN of using the BDeu score. (b) The CPT of using the BDeu score.

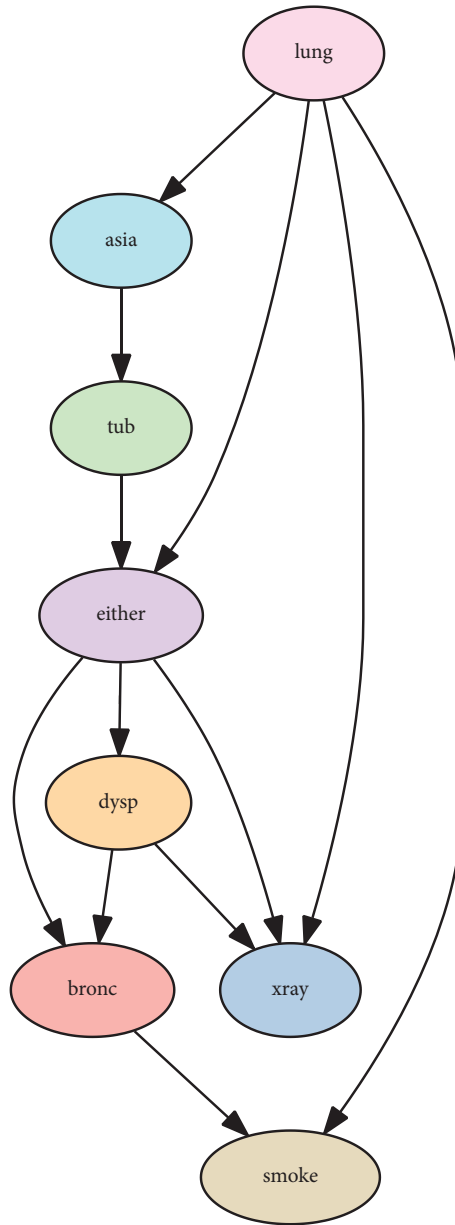
score and HD of the final ensemble learning BN is 0.45 and 9, respectively. These two evaluation indicators show that the EF-BNSL algorithm is better than the BDeu score.

Because of the randomness of the data and the uncertainty of the search process, BN structure learning results can be varied. Therefore, to improve the reliability of the experimental results, the above experimental procedure was repeated 10 times, and the results obtained by calculating the HD and F_1 are shown in Table 4. Finally, the average of HD and F_1 was calculated for 10 times of results.

It is obvious from Table 4 that the BN obtained using the EF-BNSL algorithm is larger in F_1 score than the BN obtained directly using the BDeu score. The BN obtained using the EF-BNSL algorithm is smaller in HD than the BN obtained directly using the BDeu score. The EF-BNSL algorithm on the Asia network outperforms the algorithm that uses the BDeu scoring function directly and is closer to the original standard Asia network.

Through the previous experimental method for three different types of standard networks, including small networks (Asia and Sachs), medium networks (Alarm and Insurance), and large networks (Hailfinder and Hepar2), the comparison results of six standard networks are shown in Table 5.

From the experimental results, it can be seen that the EF-BNSL algorithm proposed in this paper achieves better learning performance than the algorithm that directly using the BDeu score. The EF-BNSL algorithm does not show a greater advantage when the samples are small. Because of the lack of information in small sample data, it is difficult to mine more information of association rules. When the sample size is larger, the EF-BNSL algorithm has a better learning effect. More association rules can be mined and the black and white list can be more accurate. By using the EF-BNSL algorithm, there is a significant performance improvement in Asia and Sachs for the small size networks, and



(a)

FIGURE 5: Continued.

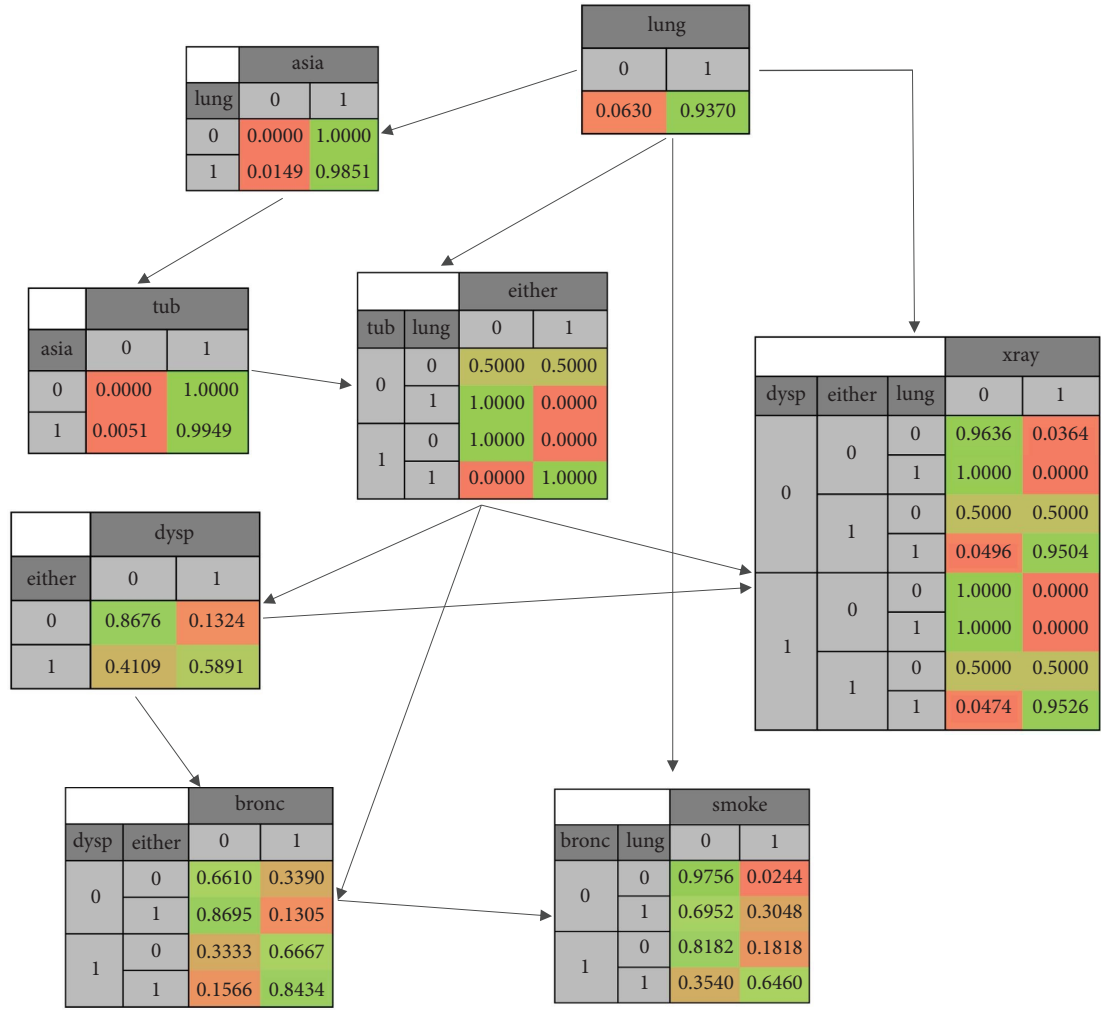


FIGURE 5: (a) The BN of using the improved scoring function. (b) The CPT of using the improved scoring function.

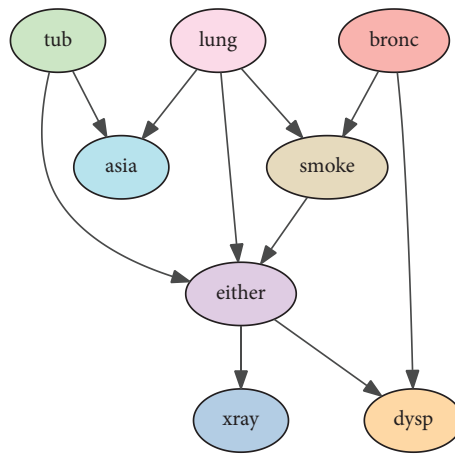


FIGURE 6: Continued.

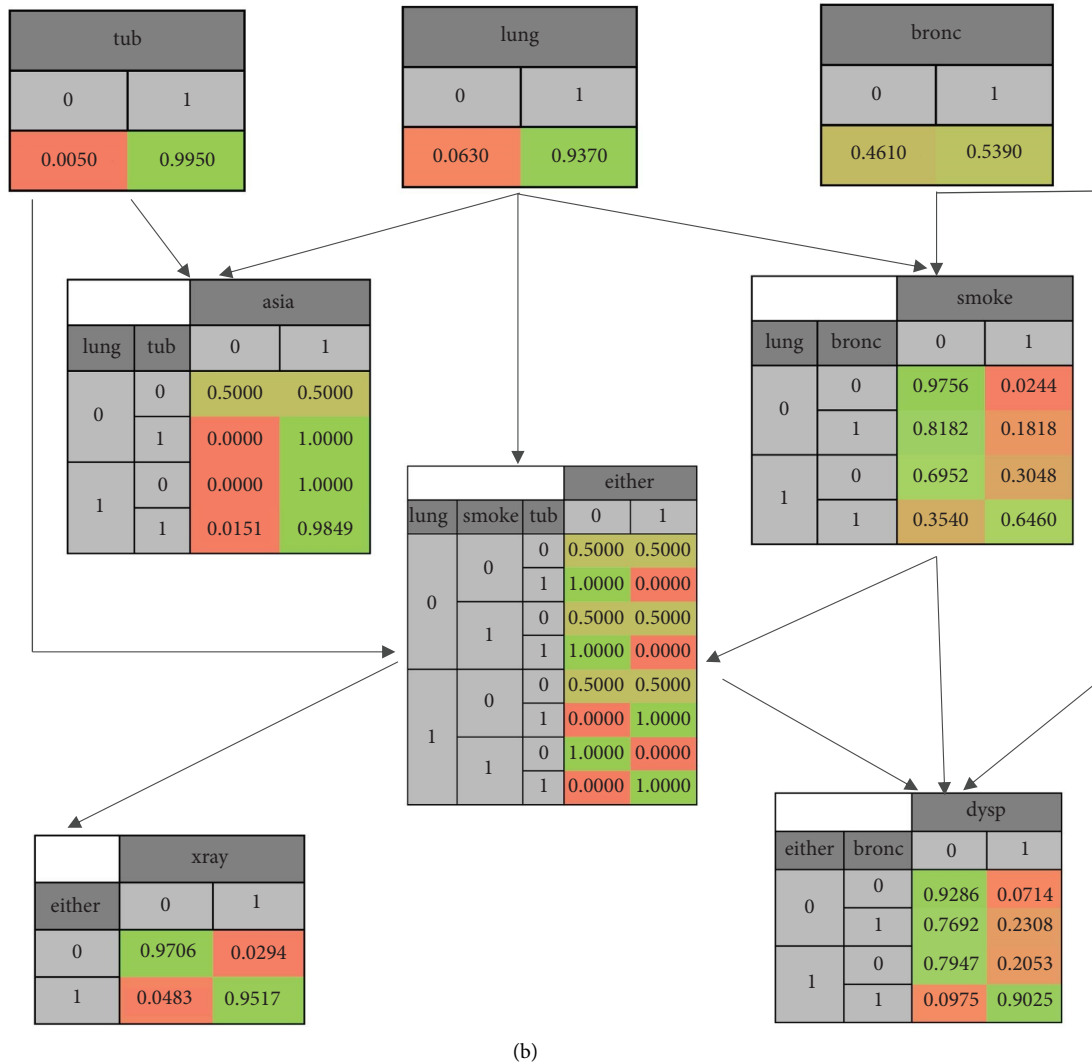


FIGURE 6: (a) Ensemble learning BN. (b) The CPT of the ensemble learning BN.

TABLE 4: Comparison of 10 results from the Asia network.

EF-BNSL		BDeu	
F_1	HD	F_1	HD
0.47	4	0.28	15
0.51	5	0.26	10
0.48	6	0.24	16
0.58	4	0.45	11
0.48	5	0.33	16
0.52	4	0.34	14
0.49	5	0.36	14
0.53	4	0.38	8
0.46	6	0.38	9
0.52	5	0.17	14

no obvious progress in Alarm and Insurance networks. In the experiment, the number of edges in the white list is much smaller than the number of edges in the black list as the number of nodes increases, which leads to a much reduced

scoring function. In future research, different weights can be further set for the edges in the black and white list. In addition, when there are more network nodes, the network search space increases exponentially. To balance the search

TABLE 5: The comparison results of six standard networks.

Sample size	Asia			Sachs			Alarm			Insurance			Hailfinder			Hepar2							
	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD					
	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_1					
1000	0.51	0.42	9.8	0.41	4.8	0.36	8.3	0.47	49.4	0.48	51.7	0.31	58.5	0.31	63.2	0.47	57.2	0.46	62.5	0.47	154.8	0.36	158.6
2000	0.53	0.43	9.2	0.49	5	0.42	7.2	0.51	45.5	0.50	47.6	0.36	53.2	0.46	55.5	0.51	61.4	0.53	65.2	0.51	128.9	0.45	130.2
5000	0.58	0.54	8.8	0.52	6	0.48	6.1	0.48	38.7	0.53	44.1	0.43	45.3	0.54	44.7	0.64	55.1	0.61	59.8	0.48	101.3	0.55	103.2

time and memory usage, the experiment adjusted down the weight of the black and white list penalty terms for large networks, which also affects the experimental results.

5. Conclusion and Future Research

In this paper, we combined the ideas of frequent item mining and ensemble learning into BN structure learning and proposed the EF-BNSL algorithm. We conducted experiments on six BNs of three types such as small, medium, and large. We used two evaluation metrics, F_1 score and HD. The results show that the EF-BNSL algorithm can effectively improve F_1 score, reduce HD, and learn the network structure that is closer to the real network.

In the experiment, as the number of nodes increases, there are more edges in the black list than in the white list. In fact, it is possible that the edges in the white list would be more important, so assigning different weights to the black and white list will be the next highlight of future research. In addition, when there are more network nodes, the network search space increases exponentially. Future research could consider using distributed computing to improve efficiency. Ensemble learning of different network structure learning algorithms, and making full use of the advantages of different algorithms for BN structure learning will be the focus of the future research.

Data Availability

Six standard networks can be downloaded from the Bayesian Network Repository (<https://www.bnlearn.com/bnrepository/>).

Conflicts of Interest

The authors declare that have no conflicts of interest.

Authors' Contributions

Guoxin Cao contributed to the conception or design of the work. Guoxin Cao and Haomin Zhang contributed to the acquisition, analysis, or interpretation of the data and drafted the manuscript. All authors reviewed the manuscript.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (61763008, 71762008, and 62166015) and the Guangxi Science and Technology Planning Project (2018GXNSFAA294131 and 2018GXNSFAA050005).

References

- [1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, Massachusetts, USA, 1988.
- [2] C. Liang, M. Ghazel, O. Cazier, and E. M. El-Kourssi, "Risk analysis on level crossings using a causal Bayesian network based approach," *Transportation Research Procedia*, vol. 25, pp. 2167–2181, 2017.
- [3] A. Gharaei, S. A. Hoseini Shekarabi, and M. Karimi, "Optimal lot-sizing of an integrated EPQ model with partial backorders and re-workable products: an outer approximation," *International Journal of Systems Science: Operations & Logistics*, pp. 1–17, 2021.
- [4] Y. Liu and S. Jin, "Application of Bayesian networks for diagnostics in the assembly process by considering small measurement data sets," *International Journal of Advanced Manufacturing Technology*, vol. 65, no. 9-12, pp. 1229–1237, 2013.
- [5] R. L. C. Souza, A. Ghasemi, A. Saif, and A. Gharaei, "Robust job-shop scheduling under deterministic and stochastic unavailability constraints due to preventive and corrective maintenance," *Computers & Industrial Engineering*, vol. 168, Article ID 108130, 2022.
- [6] A. Gharaei, A. Amjadian, and A. Shavandi, "An integrated reliable four-level supply chain with multi-stage products under shortage and stochastic constraints," *International Journal of Systems Science: Operations & Logistics*, pp. 1–22, 2021.
- [7] A. Amjadian and A. Gharaei, "An integrated reliable five-level closed-loop supply chain with multi-stage products under quality control and green policies: generalised outer approximation with exact penalty," *International Journal of Systems Science: Operations & Logistics*, vol. 9, no. 3, pp. 429–449, 2022.
- [8] A. A. Taleizadeh, A. Z. Safaei, and A. Bhattacharya, "Online peer-to-peer lending platform and supply chain finance decisions and strategies," *Annals of Operations Research*, vol. 315, pp. 1–31, 2022.
- [9] B. Nistal-Nuño, "Tutorial of the probabilistic methods Bayesian networks and influence diagrams applied to medicine," *Journal of Evidence-Based Medicine*, vol. 11, no. 2, pp. 112–124, 2018.
- [10] B. Drury, J. Valverde-Rebaza, M. F. Moura, and A de Andrade Lopes, "A survey of the applications of Bayesian networks in agriculture," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 29–42, 2017.
- [11] A. Gharaei, C. Diallo, and U. Venkatadri, "Optimal economic growing quantity for reproductive farmed animals under profitable by-products and carbon emission considerations," *Journal of Cleaner Production*, vol. 374, Article ID 133849, 2022.
- [12] A. Gharaei and E. Almehdawe, "Optimal sustainable order quantities for growing items," *Journal of Cleaner Production*, vol. 307, Article ID 127216, 2021.
- [13] A. Gharaei, A. Amjadian, A. Amjadian et al., "An integrated lot-sizing policy for the inventory management of constrained multi-level supply chains: null-space method," *International Journal of Systems Science: Operations & Logistics*, pp. 1–14, 2022.
- [14] D. M. Chickering, "Learning Bayesian networks is NP-complete," *Learning from Data*, pp. 121–130, Springer, New York, NY, UAS, 1996.
- [15] P. Spirtes and C. Meek, "Learning Bayesian networks with discrete variables from data," *KDD*, vol. 1, pp. 294–299, 1995.
- [16] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, Prediction, and Search*, MIT press, Cambridge, Massachusetts, USA, 2000.
- [17] X. Qi, X. Fan, Y. Gao, and Y. Liu, "Learning Bayesian network structures using weakest mutual-information-first strategy," *International Journal of Approximate Reasoning*, vol. 114, pp. 84–98, 2019.

- [18] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.
- [19] C. Lee and P. Beek, "Metaheuristics for score-and-search Bayesian network structure learning," in *Proceedings of the Canadian Conference on Artificial Intelligence*, pp. 129–141, Springer, Edmonton, AB, Canada, May 2017.
- [20] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Annals of Statistics*, vol. 11, no. 2, pp. 416–431, 1983.
- [21] H. Akaike, "Information theory and an extension of the maximum likelihood principle," *Selected Papers of Hirotugu Akaike*, pp. 199–213, Springer, New York, NY, USA, 1998.
- [22] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.
- [23] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: the combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995.
- [24] J. R. Alcobé, "Incremental hill-climbing search applied to Bayesian network structure learning," in *Proceedings of the First International Workshop on Knowledge Discovery in Data Streams*, Hammamet, Tunisia, April 2004.
- [25] L. Zhang and J. Zhang, "Structure learning of BN based on improved genetic algorithm," *Computer Systems and Applications*, vol. 20, no. 9, pp. 68–72, 2011.
- [26] M. Singh and M. Valtorta, "Construction of Bayesian network structures from data: a brief survey and an efficient algorithm," *International Journal of Approximate Reasoning*, vol. 12, no. 2, pp. 111–131, 1995.
- [27] J. I. Alonso-Barba, L. delaOssa, J. A. Gámez, and J. M. Puerta, "Scaling up the Greedy Equivalence Search algorithm by constraining the search space of equivalence classes," *International Journal of Approximate Reasoning*, vol. 54, no. 4, pp. 429–451, 2013.
- [28] R. Eggeling, J. Viinikka, and A. Vuoksenmaa, "On structure priors for learning Bayesian networks," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1687–1695, PMLR, Okinawa, Japan, April 2019.
- [29] Z. Wang, X. Gao, Y. Yang, X. Tan, and D. Chen, "Learning Bayesian networks based on order graph with ancestral constraints," *Knowledge-Based Systems*, vol. 211, Article ID 106515, 2021.
- [30] H. Xiao, K. Yu, and H. Wang, "The method of learning Bayesian networks combining association rules with knowledge," *Microelectronics & Computer*, vol. 25, no. 12, p. 7072, 2008.
- [31] B. Sun, Y. Zhou, J. Wang, and W. Zhang, "A new PC-PSO algorithm for Bayesian network structure learning with structure priors," *Expert Systems with Applications*, vol. 184, Article ID 115237, 2021.
- [32] X. Li, S. Zhou, J. Zhang, Z. Zhou, and Q. Xiong, "BNSL-FIM: Bayesian network structure learning algorithm based on frequent item mining," *Animals: An Open Access Journal from MDPI*, vol. 11, no. 12, p. 3475, 2021.
- [33] S. Wang and B. Qin, "Bayesian network structure learning by ensemble learning and feedback strategy," *Journal of Computers*, vol. 44, no. 6, pp. 1051–1063, 2021.
- [34] H. Baradaran Rezaei, A. Amjadian, and M. V. Sebt, "An ensemble method of the machine learning to prognosticate the gastric cancer," *Annals of Operations Research*, pp. 1–42, 2022.
- [35] R. Askari, M. V. Sebt, and A. Amjadian, "A multi-product EPQ model for defective production and inspection with single machine, and operational constraints: stochastic programming approach," in *Proceedings of the International conference on logistics and supply chain management*, pp. 161–193, Springer, Tehran, Iran, December 2020.
- [36] M. Scutari, "Bayesian network repository," 2022, <https://www.bnlearn.com/bnrepository/>.