Bayesian Network Learning Based on Characteristic Confidence Guidance Under Large Data Sets

Cai Yang

***Abstract*—**At present, the accuracy of many algorithms for Bayesian network learning under large data sets is not high. In order to solve this problem, a Bayesian network structure learning algorithm for the feature confidence guidance under the large data sets is proposed. The algorithm uses the distributed learning and the incremental learning method. At the same time, the improved SEM algorithm is used to fill the missing data, enhance the accuracy of each batch of data learning, improve the quality of the final network model. The experimental results show that the proposed algorithm has better quality of learning results, and solves the problem of insufficient memory space. The experiment of network traffic prediction shows that the proposed algorithm has a high accuracy rate of classification prediction.

***Keywords*—**bayesian network learning, feature confidence guidance, incremental learning, distributed learning.

I. INTRODUCTION

With the development of data collection technology and data analysis technology, the number of data in the world is increasing rapidly, especially the unstructured data, including voice, video, image and natural language text. This situation brings great challenges to traditional data mining algorithms and implementation schemes. The traditional data learning method is not suitable for existing large-scale data sets. Bayesian network can deal with the problem of big data well, and provides a causal information processing method, which is widely applied in industrial control and artificial intelligence.

In recent years, many positive efforts have been made in the classification prediction and the discovery of potential causal relationship, and the corresponding progress has been made. Anders L.Madsen et al. describe a new approach to parallelization of the (conditional) independence testing as experiments illustrate that is by far the most time consuming step. The proposed parallel PC algorithm is evaluated on data sets generated at random from five different real-world Bayesian networks [1]. Marco Benjumeda et al. show how information about the most common queries of multidimensional Bayesian network classifiers affects the complexity of these models. The upper bounds are provided for the complexity of the most probable explanations and marginals of class variables conditioned to an instantiation of all feature variables [2]. This paper shows why the consideration of data distribution can yield a more effective similarity measure. In addition, the current work both introduces a new scalable similarity measure based on the posterior distribution of data and develops a practical algorithm that learns the proposed measure from the data [3]. Santiago Cortijo et al has introduced an alternative model called a ctdBN that lies in between. It is composed of a "discrete" Bayesian network (BN) combined with a set of univariate conditional truncated densities modeling the uncertainty over the continuous random variables given their discrete counterpart resulting from a discretization process [4]. Enrique Castillo et al present several new and original contributions to complement the inference engine tools of these models to provide new and relevant information about safety and backward analysis on one hand, and to learn the complex multidimensional joint probabilities of all variables, on the other hand [5]. The authors study prospects of representing relationships between variable groups using Bayesian network structures. They show that for dependency structures between groups to be expressible exactly, the data have to satisfy the so-called groupwise faithfulness assumption [6].

These algorithms propose different Bayesian network learning schemes, but the accuracy of learning under large data sets is not high. In order to solve this problem, a Bayesian network structure learning algorithm for the feature confidence guidance under the large data sets(FCLDS-LBN) is proposed. This algorithm is based on distributed learning, and proposes an incremental learning scheme. And it can dynamically update the model by the new observation data, so as to reduce the calculation cost as more as possible. Firstly, the training data is divided into small blocks. Secondly, multiple Bayesian network subnets are obtained by using block data. Finally, these subnets are classified and predicted by the Boosting method. The experiment shows that the algorithm is beneficial to learn the Bayesian network with better fitting degree. At the same time, the training process of Bayesian network is accelerated, and the higher accuracy of classification prediction is ensured.

This work was supported in part by the cooperative education project of the Ministry of education of the People's Republic of China (201701004011).

Cai Yang is with College of Computer and Information Technology, Nanyang Normal University, Nanyang 473061, Henan, China (corresponding author; e-mail: nyyc@163.com).

II. Basic Knowledge

A. Bayesian network

The Bayesian network(BN) is also called the reliability network, which is composed of a Directed Acylic Graph (DAG) and a conditional probability table (CPT) [7].

The BN model of n random variable $X=\{X_1,X_2,...,X_n\}$ is a two tuple, expressed as $B= (B_s, B_p)$. $B_s= (X, E)$ is a directed acyclic graph. Among them, $X=\{X_1,X_2,...,X_n\}$ are node sets, and each node can be regarded as a variable that takes discrete or continuous values [8]. E is a set of directed edges. Each edge represents a dependency between two nodes. The degree of dependency is determined by conditional probability parameters. B_s is called the network structure of BN.

$$B_p = \left\{P(X_i \left| \prod X_i \right.), X_i \in X \right\} \quad (1)$$

In (1), B_p is a set of conditional probability distributions of Bayesian network models. $\prod X_i$ is the set of all parent nodes of B_s in X_i, which represents the conditional probability distribution of node X_i under the condition of a certain value combination of its parent node. This shows that in Bayesian network models, the value of nodes depends on the value state of their parent nodes [9]. The problem of learning Bayesian network is described as: given a set of training instances set $D=\{d_1, d_2,..., d_n\}$, find a match with the best network B. In this way, learning Bayesian network problem is transformed into optimization problem.

The real solution to this problem is that heuristic search is carried out in the space of possible network formation. The key step of search success is that a reasonable scoring function is found to guide the search of various network structures [10]. Thus, an optimal network with the highest matching degree of training data is obtained.

B. Learning of Bayesian networks under large data sets

For the Bayesian network learning under large data sets, the general solutions are the batch learning of the large scale data sets. There are two common solutions: simple incremental learning and maximum posterior probability increment learning [11].

The first great data set to study the BN learning is Friedman, which adds new research to the study of BN. Because of the large amount of data, it can not be read into memory at all times, and only incremental learning is carried out in batches. However, if the data is simply divided into several blocks and the information learned is not well preserved, the results are often very difficult to satisfy. In the process of incremental learning, data can not be directly applied to the classical network structure scoring method because the data come in batches. More flexible ways should be taken to deal with it.

C. General strategy of incremental learning in BN

The most common strategy for incremental learning is that data is divided into a number of batches to learn, and each batch needs find a network with the greatest posterior probability as the initial network for the next batch of data learning [12].

The advantage of this algorithm is that each space cost is stable and reasonable, because it only saves the current batch data, and the data that has been learned is completely abandoned. Its disadvantage is that the network will be locked on a network model and lose its adaptability to new data after several iterations. Therefore, a batch of candidate networks generated by the best network currently learned are used as a priori network for the next batch of data learning. At the same time, a full statistic is introduced to preserve the information that has been learned, so that the network can save more prior knowledge and make the later learning network better fit with the potential network model. When each batch of data arrives, the current batch data is used to update the full statistics. Each search must find a network with the highest score from the candidate network, and then iterate in order until the algorithm converges.

D. SEM Algorithm

Data deletion is a common phenomenon in Bayesian network learning, especially for large datasets. In general, the missing data is processed by incomplete data, and then Bayesian network learning is done on the complete dataset. The representative algorithm is the SEM (Structure Expectation-Maximization) algorithm. The specific work flow is as follows: define $B_i= (G_i, \theta_i)$ and assign i=0 to represent the initial state of Bayesian network. After the K iteration, the optimal network $B_k= (G_k, \theta_k)$ is obtained. The steps of the (k + 1) iteration are as follows:

Step 1: According to the current optimal network B_k , the data set D_i is complementing by using the EM algorithm (Expectation Maximization algorithm). Finally, the completed data set D_{it} is obtained.

Step 2: According to the complete dataset D_{it}, the network structure is optimized to get B_{k+1}. It is expressed in formula (2).

$$B_{k+1} = (G_{k+1}, \theta_{k+1}) \quad (2)$$

The SEM algorithm selects the Bayesian network with the highest expected score at every step of optimization. According to the best network, the SEM algorithm uses the EM algorithm to complement the complete set of data, and obtains the expected statistical factors required by the statistics. Under some assumptions, the score function can be decomposed, and the problem of data missing learning is transformed into a complete data set learning Bayesian network. The algorithm tries to converge after successive iterations on the structure and parameters of the network. To some extent, a solution to the problem is proposed, but the high probability converges to the local optimal.

III. Designment Of The Algorithm

A. Distributed learning design scheme

There are many ways to learn Bayesian network structure [13]. In order to improve the quality and efficiency of the algo-

rithm, the distributed algorithm structure is used int the FCLDS-LBN algorithm. Its process is divided into three stages.

Step 1: The large data D is evenly divided into n blocks. Expression is expressed as: $D=\{D_1, D_2,..., D_n\}$. The D set is the input of the training stage, and the MMHC (Max-Min Hill Climbing) algorithm is executed in parallel to learn the Bayesian network structure. The learning network structure is represented as G_i^*.

Step 2: According to the MapReduce framework, the parallel computing power of MapReduce, the training results of the block data can be obtained in a short time. This result is expressed as: $D_i=\{N_1, N_2,..., N_n\}$. Then the network structure of the subset is derived.

Step 3: This stage is mainly the relearning of the subnet. The prediction results of subnet are classified two times. The set of coefficients is introduced, which is represented as: $\{a_1,a_2,\ldots,a_n\}$. Then the weight of the prediction results of each subnet is adjusted. Finally, the final learning results are obtained, which are expressed as $D_i^*=\{r_1, r_2,..., r_n\}$.

B. Incremental learning scheme

Increase the amount of learning is a kind of online learning. In each iteration, new data samples are received and historical data are adjusted. The incremental learning process of the FCLDS-LBN algorithm is shown in Fig.1.

As you can see in Figure 1, in the process of model search, the search boundary of a Bayesian network G is maintained. It includes a number of networks that can obtain higher scores in the designated scoring mechanism. The data set of the Bayesian network structure is obtained, which is represented as S.

Then, the newly observed data sample D_i is received by the collection of S. Next, the FCLDS-LBN algorithm is called to find the edge of the high confidence. The Bayesian network structure has been continuously updated. The optimal network is found by using the scoring mechanism of S, which is expressed as G '. At the same time, the update of the Bayesian network parameter F and the data set S is completed. According to the new parameters, the new network structure is obtained, which are expressed as: G and θ. At this point, the update of the structure has been completed [14].

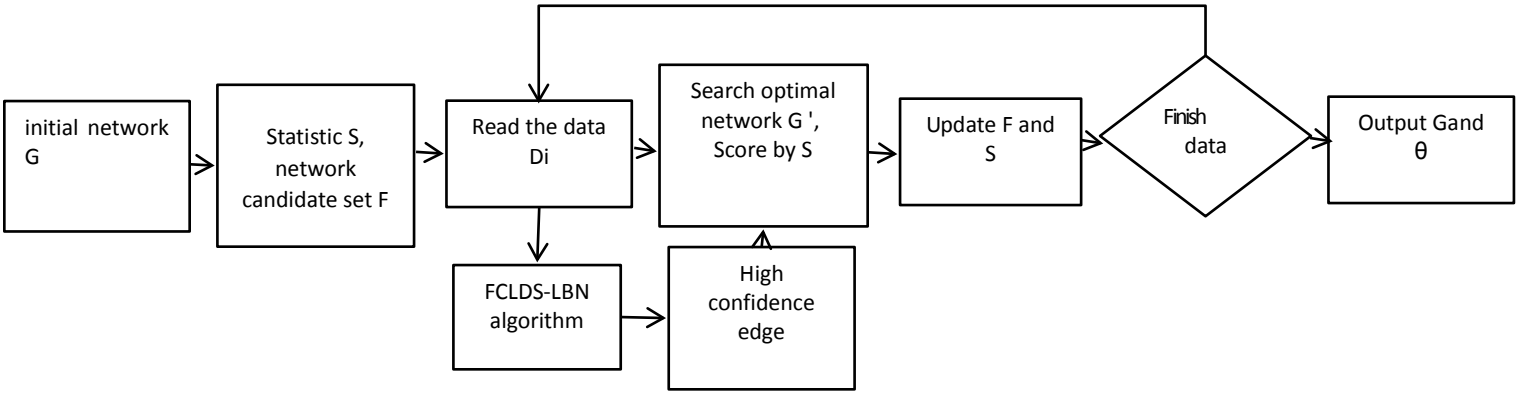

Fig.1. Incremental learning flow chart under large data

C. Learning of Bayesian networks under the guidance of characteristic confidence

The method of feature confidence guidance is integrated into the incremental learning process, which can improve the precision of data set learning and get better results. There are many confidence methods in the Bayesian network [15]. In the FCLDS-LBN algorithm, the confidence of the node sequence is used. The probability that the order of the nodes in graph G appears in the graph set G* is calculated. The variable G* is expressed as: $G^*=\{G_1, G_2, \ldots, G_m\}$. The sequential relation between nodes is represented by a three tuple, which is expressed in (3).

$$\langle u, v, \rho \rangle \tag{3}$$

In (3), the variables u and v represent two nodes. Among them, v is the successor node of u, and u is the precursor node of v. The variable ρ represent the probability of this edge <u,v> appears in graph G_i. In addition, $0 \le \rho \le 1$. When there is a sequence relationship between u and v in the graph, it is expressed as <u, v, 1.0>. Otherwise, it is represented as <u, v, 0.0>. In order to improve the learning quality of the graph, a threshold k is set. When the variable ρ in (3) is greater than k, the order of the relationship between u and v is identified in the network structure [16]. After accessing all three tuples in the DAG graph set G*, a node sequence relation set can be obtained, which is represented by the variable P. All the elements with a confidence degree greater than the threshold K are included in the set P. Equation (4) is used to calculate the confidence of the node order [17].

$$C(u,v)=\frac{1}{m}\sum_{i=1}^{m}\rho \tag{4}$$

In which, the variable m represents the number of graphs in

G*. When there is a relationship between u and v in the order of nodes in figure G_i, ρ is assigned 1.0. Otherwise, ρ is assigned 0.0. C(u, v), which is calculated by (3), represents the sequential confidence of node u and node v.

The MMHC(Max-Min Hill Climbing) algorithm uses the evaluation mechanism to select a higher network structure, and finally obtains an optimal network on the quality and structure of the score [18]. The MMHC algorithm is used to solve such problems because the search algorithm may cause local optimal. The FCLDS-LBN algorithm search strategy is as follows. Firstly, the node sequence relation set P is used to guide the MMHC algorithm and learn on each batch of data sets. Then, the network is updated by using the high confidence node sequence relation, which is learned through the MMHC algorithm. With this analogy, a better network model is finally obtained [19].

D. Generating the best subset of data

Bootstrapping is a computer simulation method that can simulate the actual situation of sampling by multiple operations. Through the distributed learning, the Bayesian network subnet is obtained to generate the best module size for the data by using the Bootstrapping method [20]. A value N_{sample} is set to indicate the number of samples with the sampled data placed back. At the same time, $N_{resample}$ represents the number of samples of the sampled data that are not placed back. The data set satisfies the (5):

$$N_s = N_{sample} + N_{resample} \tag{5}$$

A parameter α is introduced to represent the ratio of N_{sample} to N_s. That is to say, $\alpha = N_{sample}/N_s$. Then, the (5) can be written as $N_{sample} = (1-\alpha) N_s$. In this way, the size N_{sample} of the sampled data is represented as a function of the α. Through adjusting the size of α, the proportion of repeated sampling can be adjusted [21].

IV. Implementation of the FCLDS-LBN Algorithm

A. Reformative SEM algorithm

The phenomenon of data loss is more common in Bayesian network learning, especially in large data sets. In general, method of filling missing data is used to deal with this problem. Then, the whole data set is learned by Bayesian network. The SEM(Structure Expectation Maximization) algorithm is the most widely used algorithm in this field [22]. However, the execution results of this algorithm have a strong dependence on the initial parameters. Therefore, a poor initial value will lead to an increase in the number of cycles in the learning process, and reduce the time performance of the algorithm and the learning accuracy of the results [23]. In order to solve this problem, the improved SEM algorithm is proposed. The algorithm flow is as follows:

Step 1: The initial value is optimized. The following steps are included.

(1) The data set is set to D, and then the K initial parameter values are generated randomly. They are respectively expressed as: θ_1^0, θ_2^0, ..., θ_k^0. The expected value of the likelihood function of θ_i^0 is calculated by using (6).

$$L(\theta \mid \theta_i^0) = \sum_L \sum_{X_i} InP(D_L, X_L \mid \theta) P(X_L \mid D_L, \theta_i^0) \tag{6}$$

In this, the variable D_L represents the current set of data, and the variable X_L represents all the variables.

(2) The next estimated value is selected by maximizing the current expected likelihood function value. The method of calculation is shown in (7).

$$\theta_i^1 = \arg\max E\,[P(D \mid \theta) \mid D, \theta_i^0, M^0] \tag{7}$$

According to using (6), k results can be obtained.They are represented as:θ_i^1(i=1, 2,...,k). Among them, argmax () is a parameter that has the maximum score.

(3) In these results, a best value is selected according to (8).

$$\theta^0 = \arg\max Q(\theta_i^1) \quad (i = 1,2,...,k) \tag{8}$$

Step 2: An equation is defined as: $B_i = (G_i, \theta_i)$, and the variable i is assigned to 0.Among them, θ^0 is the initial parameter obtained by Equation (8), which indicates the initial state of the Bayesian network.

Step 3: The EM(Expectation Maximization) algorithm is used to generate the optimal network B_k.It is expressed in (9).

$$B_k = (G_k, \theta_k) \tag{9}$$

Accordingly, the data set D_i is modified.The modified dataset is represented as D_{it}.

Step 4: According to the complete data set D_{it}, the network structure is optimized. B_{k+1} is obtained by this.The final result is obtained by analogy.

B. Designment of the FCLDS-LBN algorithm

The algorithm is described as follows.

Input: D data sets (batch input), said: D= {D1, D2,... }. Threshold K is taken as follows: 1.0, 0.9, 0.8, 0.7.

Output: Bayesian network learning model.

Step 1: The data subset D_i is read.

Step 2: In the data subset D_i, k data sets are extracted by distributed learning method. It is expressed as: $D_i^* = \{D_i^1, D_i^2, ... D_i^k\}$.

Step 3: According to the MMHC algorithm, the higher quality Bayesian network G_{ij} is learned on the D_{ij} data set.

Step 4:

while Traversing all the graphs in G_{ij}^*

  while Traversing all three tuples in the G_{ij}

    Calculate C (u, v) through (4).

  end while

end while

Step 5: The node three tuples greater than the threshold k are recorded in the node sequence relationship set P.

Step 6: Combined with the MMHC algorithm, the set P is used to guide the learning of Bayesian networks in the data subset D_i.

Step 7: The final network model B is obtained.

The step 1 to step 5 in the algorithm are sampled for each batch of data, and the node sequence relation set P is obtained. The six step is Bayesian learning of the sequential confidence of the data [24]. After each batch of data is completed, the optimal network structure B is used to update the candidate network F.

The time consumption of the FCLDS-LBN algorithm is mainly the search process of MMHC and the sequential confidence of the nodes. The time complexity of the MMHC algorithm is expressed as: I_{max}*O(n + e). In this, I_{max} represents the number of iterations, and n and e represent the number of nodes and the number of edges. The time complexity of the search process of node sequence confidence is O (k*r). Where k is the number of the middle graphs, and r is the number of nodes in the graph. That is to say, the time complexity of the FCLDS-LBN algorithm is expressed as: I_{max}*(O (n+e)+ O (n+r)). It can be seen that the storage space has increased. However, a slight increase in storage space for better solutions is acceptable in most cases [25].

V. Verification of the Algorithm

The hardware environment used in the experiment is Lenovo-RD650, and the specific configuration is as follows:

CPU model: Xeon E5-2650 V3.

Memory capacity: 128G.

Operating system: Linux.

The data used for the test comes from http://www.norsys.com. According to the probability distribution map of ALARM network, the experimental data sets are obtained, which are 10 groups. Each group contains a set of training data sets with 10000 records and a test data set of 1000 records. At the same time, in order to test the processing ability of abnormal data, the data sets used in the training are randomly generated from abnormal data, about 5% to 10%. The result of the experiment is taken the average value of each set of data sets [26]. The K2 algorithm and the IBN - M algorithm are very widely used in practical applications. In order to test the actual effect of the algorithm, these two algorithms are used for comparison.

A. Comparison of data fitting degree

The network model is analyzed from the view point of data fitting. The test is represented by the relative value. The contrast results are shown in Table 1.

Table 1. Comparison table of data fitting degree

Algorithm	K2	IBN-M	FCLDS-LBN			
Confidence level	-	-	0.7	0.8	0.9	1.0
1	-4.7258	-4.7369	-4.6395	-4.6251	-4.6192	-4.6426
2	-4.7223	-4.7205	-4.6261	-4.6193	-4.6128	-4.6186
3	-4.5739	-4.6337	-4.6235	-4.6128	-4.5905	-4.6151
4	-4.6103	-4.6209	-4.6021	-4.6032	-4.5562	-4.5925
5	-4.6012	-4.6112	-4.6028	-4.5728	-4.5325	-4.5905
6	-4.5663	-4.5698	-4.5762	-4.5605	-4.5213	-4.5569
7	-4.5623	-4.5591	-4.5411	-4.5412	-4.5105	-4.5256
8	-4.5312	-4.5523	-4.5423	-4.5368	-4.5021	-4.5351
9	-4.5418	-4.5426	-4.5402	-4.5289	-4.4962	-4.5262
10	-4.5271	-4.5293	-4.5363	-4.5318	-4.4921	-4.5198

As can be seen from Table 1, in general, the FCLDS-LBN algorithm is better than the other two algorithms in data fitting degress. When the confidence level is 0.9, the data fitting degree of the FCLDS-LBN algorithm is -4.4921, which is greater than the result of K2 algorithm, and is also larger than that of IBN-M algorithm. This shows that the network model learned by the FCLDS-LBN algorithm is better than the K2 algorithm and the IBN-M algorithm in data fitting [27].

B. Performance comparison

The performance of the three algorithms is tested, and the results of the experiment take the average of each set of data sets [28]. The comparison results are shown in Fig.2.

As can be seen from Figure 2, the accuracy of the three algorithms is not high when the number of samples is 500. The reason is that when the number of samples is less, it is difficult to determine the dependence between attributes through a data

set [29]. As the number of samples increases, the accuracy of the K2 algorithm is gradually improved. The IBN-M algorithm is unstable. When the number of samples is 2000, 3000 and 4000, the accuracy is lower than the K2 algorithm.

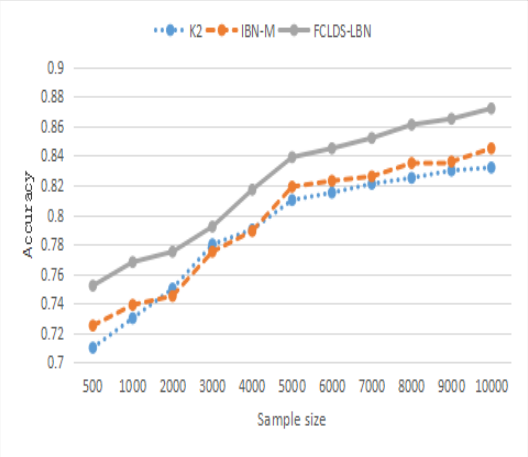

Fig.2. Performance comparison of algorithms

In other cases, the number of samples is higher than the K2 algorithm. The FCLDS-LBN algorithm, compared with the other two algorithms, has a steady improvement in accuracy. Therefore, the FCLDS-LBN algorithm can get a more accurate network structure.

C. Comparison of network learning results

The results of the three algorithms are compared. The number of sampling times is n=100, and the confidence is k={0.7, 0.8, 0.9, 1.0}. The results are shown in Table 2.

It can be seen from Table 2 that the FCLDS-LBN algorithm learns 43, 46, 52, and 46 copies three tuples at confidence levels 0.7, 0.8, 0.9, and 1.0 when the data is completed.

Table 2. Comparison of network structure obtained by learning

Algorithm	Confidence level	1	2	3	4	5	6	7	8	9	10
K2	-	26	29	28	26	28	28	32	33	35	36
IBN-M	-	26	27	25	26	27	31	31	35	37	35
FCLDS-LBN	0.7	30	32	37	36	42	41	42	43	45	43
	0.8	31	36	41	38	45	46	45	45	47	46
	0.9	36	37	40	44	46	46	48	51	51	52
	1.0	31	34	36	39	40	42	45	46	47	46

At this point, the K2 algorithm learned 36 three tuples, while the IBN-M algorithm learned 35 three tuples. This shows that the FCLDS-LBN algorithm has higher network architecture than the K2 algorithm and the IBN-M algorithm [30]. In terms of convergence speed, the FCLDS-LBN algorithm, which has node sequence confidence, can find better quality network faster, while IBN-M algorithm and K2 algorithm are inferior.

D. Storage space contrast

The storage consumption space is compared as shown in Fig. 3. It can be seen from Fig. 3 that the FCLDS-LBN algorithm takes up a moderate amount of memory. In comparison, the IBN-M algorithm takes up less memory. The reason is that the FCLDS-LBN algorithm includes the node sequence confidence. The FCLDS-LBN algorithm takes a little less space than the K2 algorithm, the reason is that the K2 algorithm uses the greedy search processing model. After the learning of some batch data, the memory space is not up and down.

Although the proposed algorithm accounts have more memory space, it is within the acceptable range.

To sum up, the FCLDS-LBN algorithm is excellent in data fitting degree, algorithm performance, network structure learning and data analysis and prediction ability [32]. Furthermore, the FCLDS-LBN algorithm pays more attention to the quality and accuracy of each set of subsets of data. This helps to eventually build a better network of quality.

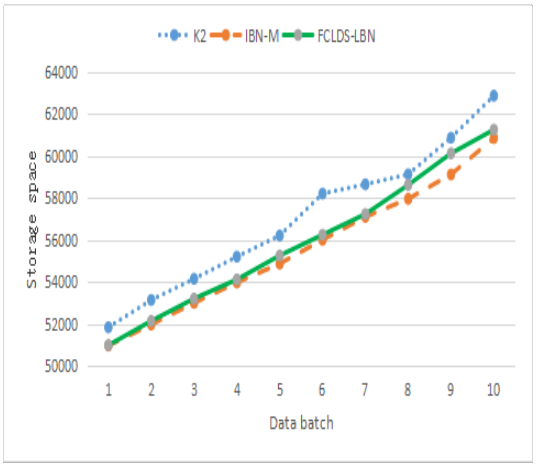

Fig.3. Storage space comparison

VI. Application of the Algorithm

A. Test case

In order to test the analysis ability of the algorithm on the data set, the three algorithms are compared from the aspect of prediction accuracy [31]. The gateway log files of the school network center are analyzed. The data is counted every 5 minutes, and the current time are written in the log at the same time. Every day, about 3900000 data are written to the log file. The occupied storage space is about 810M. The data used in the test is 5 days' log information. The network access traffic is predicted based on the network structure and parameters.

The data recorded in the log follow the rules:

Rule 1：SSL encryption technology is used to encrypt user ID and user IP address.

Rule 2：the user's IP address is represented by a string of 4 decimal digits.

Rule 3：the number of bytes is used to express network traffic.

B. Test result

In order to test the ability of the algorithm to analyze the data set, the prediction of the three algorithms is compared by the prediction accuracy. The test results are shown as Table 3. From table 3, it can be seen that the K2 algorithm has 5 days of traffic prediction, and the accuracy rates are 51.26%, 56.37%, 57.68%, 56.85% and 57.26% respectively. The IBN-M algorithm is used to predict the traffic flow for 5 days. The accuracy rate is 52.93%, 61.28%, 60.75%, 61.83% and 61.59%. In sharp contrast, the FCLDS-LBN algorithm also performs 5 days of traffic prediction, with an accuracy rate of 62.98%, 65.13%, 68.39%, 69.86% and 70.95%. Obviously, the prediction accuracy of FCLDS-LBN algorithm is the highest, and the accuracy of prediction can reach over 70%. Moreover, with the increase of data volume, the prediction accuracy is higher. This is because the data learned have been used to predict network traffic.

Table 3. Comparison of prediction accuracy of network access traffic

	First day	Second day	Third day	Forth day	Fifth Day
K2	51.26%	56.37%	57.68%	56.85%	57.26%
IBN-M	52.93%	61.28%	60.75%	61.83%	61.59%
FCLDS-LBN	62.98%	65.13%	68.39%	69.86%	70.95%

It also shows that in the case of large scale data sets, the performance of FCLDS-LBN algorithm in Bayesian network learning is valid, and it has certain guiding significance for practical application. At the same time, the experimental results also show that in the aspect of Bayesian network learning, data should inspire each other and complement each other, so that a better network model can be obtained.

VII. Conclusion

In order to solve the problem of Bayesian network learning in large data sets, the FCLDS-LBN algorithm is proposed. First, the confidence-directed learning strategy is integrated into incremental learning. It enhances the learning accuracy of data structure of every batch data under large data sets, guarantees the quality of learning, and reduces the final network accuracy caused by cumulative error margin. Then, the improved SEM algorithm is used to complete missing data. The efficiency and the precision of learning are improved. The experimental results show that the FCLDS-LBN algorithm is superior to the K2 algorithm and the IBN-M algorithm in the network structure, the data fitting degree and the data analysis and prediction ability. The FCLDS-LBN algorithm can learn a relatively accurate network model, and the result of learning is more accurate. The result of network traffic prediction shows that the proposed algorithm has a good use value. In the learning process of Bayesian network in large data sets, some statistics are thrown away as the network structure changes. And the statistics thrown aside will still have an impact on the subsequent Bayesian network learning. This is the next step that needs further study.

References

[1] Anders L.Madsenab, Frank Jensena, Antonio Salmerón, Helge Langseth, and Thomas D. Nielsen, "A parallel algorithm for Bayesian network structure learning from large data sets", *Knowledge-Based Systems*, vol.117, no.1, pp. 46-55, 2017.

[2] Marco Benjumeda, Concha Bielza, and Pedro Larrañaga, "Tractability of most probable explanations in multidimensional Bayesian network classifiers", *International Journal of Approximate Reasoning*, vol.93, pp. 74-87, 2018.

[3] Davood Zabihzadeh, Reza Monsefi, and Hadi Sadoghi Yazdi, "Sparse Bayesian similarity learning based on posterior distribution of data", *Engineering Applications of Artificial Intelligence*, vol.67, pp. 173- 186, 2018.

[4] Santiago Cortijo, Christophe Gonzales, "On conditional truncated densities Bayesian networks", *International Journal of Approximate Reasoning*, vol.92, pp.155-174,2018.

[5] Enrique Castillo, Zacarías Grande, Elena Mora, Xiangdong Xu, and Hong K. Lo, "Proactive, Backward Analysis and Learning in Road Probabilistic Bayesian Network Models", *Computer -Aided Civil and Infrastructure Engineering*, vol.32, no.10, pp. 820-835, 2017.

[6] P Parviainen, S Kaski, "Learning structures of Bayesian networks for variable groups", *International Journal of Approximate Reasoning*, vol.88, pp. 110-127, 2017.

[7] RF Ropero, S Renooij, LCVD Gaag, "Discretizing environmental data for learning Bayesian-network classifiers", *Ecological Modelling*, vol.368, pp. 391-403, 2018.

[8] C Contaldi, F Vafaee, PC Nelson, "Bayesian network hybrid learning using an elite-guided genetic algorithm", *Artificial Intelligence Review*, vol.293, pp. 1-28, 2018.

[9] S Nie, M Zheng, Q Ji, "The Deep Regression Bayesian Network and Its Applications: Probabilistic Deep Learning for Computer Vision", *IEEE Signal Processing Magazine*, vol.35, no.1, pp. 101-111, 2018.

[10] H Ramchoun, M Ettaouil, "Hamiltonian Monte Carlo based on evidence framework for Bayesian learning to neural network", *Soft Computing*, vol.6, pp. 1-11, 2018.
[11] J Kwisthout, "Approximate inference in Bayesian networks: Parameterized complexity results", *International Journal of Approximate Reasoning*, vol.93, pp. 119-131, 2018.
[12] M Scanagatta, G Corani, M Zaffalon, J Yoo, U Kang, "Efficient Learning of Bounded-Treewidth Bayesian Networks from Complete and Incomplete Data Sets", *International Journal of Approximate Reasoning*, vol.95, pp. 152-166, 2018.
[13] R Sardinha, A Paes, G Zaverucha, "Revising the Structure of Bayesian Network Classifiers in the Presence of Missing Data", *Information Sciences*, vol.439-440, pp. 108-124, 2018.
[14] H Nakada, Y Ichisugi, "Context-Dependent Robust Text Recognition using Large-scale Restricted Bayesian Network", *Procedia Computer Science*, vol.123, pp. 314-320, 2018.
[15] S Kabir, M Walker, Y Papadopoulos, "Dynamic system safety analysis in HiP-HOPS with Petri Nets and Bayesian Networks", *Safety Science*, vol.105, pp. 55-70, 2018.
[16] S Mehdizadeh, AK Sales, "A Comparative Study of Autoregressive, Autoregressive Moving Average, Gene Expression Programming and Bayesian Networks for Estimating Monthly Streamflow", *Water Resources Management*, vol.15, pp. 1-22, 2018.
[17] CJ Butz, JS Oliveira, AED Santos, AL Madsen, "An empirical study of Bayesian network inference with simple propagation", *International Journal of Approximate Reasoning*, vol.92, pp. 198-211, 2018.
[18] L Cuypers, P Libin, Y Schrooten, K Theys,and VCD Maio,"Exploring resistance pathways for first-generation NS3/4A protease inhibitors boceprevir and telaprevir using Bayesian network learning", *Infection, Genetics and Evolution*, vol.53, pp. 15-23, 2017.
[19] AR Masegosa, AM Martinez, and H Langseth, "Scaling up Bayesian varia- tional inference using distributed computing clusters", *International Jour- nal of Approximate Reasoning*, vol.88, pp. 435-451, 2017.
[20] M Dimkovski, A An, "A Bayesian model for canonical circuits in the neo- cortex for parallelized and incremental learning of symbol representations", *Neurocomputing*, vol.149, no.4, pp. 1270-1279, 2015.
[21] Barbaros Yeta, Zane B.Perkinsb, and Todd E.Rasmussen, NR Tai, and DW Marsh, "Combining data and meta-analysis to build Bayesian net- works for clinical decision support", *Journal of Biomedical Informatics*, vol.52, no.C, pp. 373-385, 2014.
[22] Flávio LuizSeixasa, Bianca Zadroznyb, Jerson Laks ,Aura Conci, and Débora Christina Muchaluat Saade, "A Bayesian network decision model for supporting the diagnosis of dementia, Alzheimer's disease and mild cognitive impairment", *Computers in Biology and Medicine*, vol.51, no. 7, pp. 140-158, 2014.
[23] Hossein Amirkhani, Mohammad Rahmati, "Expectation maximization based ordering aggregation for improving the K2 structure learning algorithm", *Intelligent Data Analysis*, vol.19, no.5, pp. 1003-1018, 2015.
[24] Song Ko, Dae-Won Kim, "An efficient node ordering method using the conditional frequency for the K2 algorithm", *Pattern Recognition Letters*, vol.40, no. 4, pp. 80-87, 2014.
[25] H Wang, Y Yan, J Hua, Yutao Yang, and Xun Wang,"Pedestrian recognition in multi-camera networks using multilevel important salient feature and multicategory incremental learning", *Pattern Recognition*, vol.67, no.C, pp. 340-352, 2017.
[26] Katerine Diaz-Chito, Konstantia Georgouli, and Anastasios Koidis, "Incremental model learning for spectroscopy-based food analysis", *Chemometrics and Intelligent Laboratory Systems*, vol.167, no.4, pp. 123-131, 2017.
[27] Emrah Ergul, "Relative attribute based incremental learning for image recognition", *CAAI Transactions on Intelligence Technology*, vol.2, no.1, pp. 1-11, 2017.
[28] Won-Pyo Hong, Hon-Zong Choi, "An automatic size measuring algorithm for SEM", *International Journal of Precision Engineering and Manufacturing*, vol.16, no.7, pp. 1487-1491, 2015.
[29] Y. Slaoui, G. Nuel, "Parameter Estimation in a Hierarchical Random Intercept Model with Censored Response: An Approach using a SEM Algorithm and Gibbs Sampling", *Sankhya B*, vol.76, no.2, pp. 210-233, 2014.
[30] Ivan Piza-Davila, Guillermo Sanchez-Diaz, Manuel S. Lazo-Cortes, and Luis Rizo-Dominguez, "A CUDA-based Hill-climbing Algorithm to Find Irreducible Testors from a Training Matrix", *Pattern Recognition Letters*, vol.95, no.5, pp. 22-28, 2017.
[31] Hélder S.Sousaa, FranciscoPrieto-Castrillo, JC Matos , JM Branco, and PB Lourenã§O, "Combination of expert decision and learned based Bayesian Networks for multi-scale mechanical analysis of timber elements", *Expert Systems with Applications*, vol.93, pp. 156-168, 2018.
[32] S Chaudhary, S Indu, S Chaudhury, "Video-based road traffic monitoring and prediction using dynamic Bayesian networks", *IET Intelligent Transport Systems*, vol.12, no. 3, pp. 169-176, 2018.

Cai Yang was born on June 20, 1979**.** She received the Master's degree in computer software and theory from Northwest University of China. Currently, she is a researcher at Nanyang Normal University, China. Her major research interests include network information processing and algorithm analysis. She has published many papers in related journals. In addition, she is a member of China Computer Federation.