DYNAMIC BAYESIAN NETWORK STRUCTURE LEARNING OPTIMIZATION ALGORITHM BASED ON BLDOO+CS

Yuexia XIONG[1], Fuguo LIU[2]*, Guodong LI[3]

The Dynamic Bayesian Network, as a special form of Bayesian network, has been widely studied at present. However, due to the dynamic nature of the network, how to effectively grasp the Dynamic Bayesian Network is facing new challenges. In order to further improve the efficiency of Dynamic Bayesian Network learning algorithm, an improved algorithm for Dynamic Bayesian Network structure learning based on BLDO+CS is proposed, This algorithm integrates BIC based local dynamic optimization operators into the cuckoo algorithm, enabling real-time dynamic optimization of the network structure. The structure learning of Dynamic Bayesian Network is divided into two stages: prior network and transition network, which simplifies the learning complexity; The transfer network utilizes the BLDOO+CS algorithm to learn and modify its structure. The experiment shows that in the test of dynamic ASIA and MILDEW networks as data, HD_0, $HD_\rightarrow$ and CE of the proposed algorithm are better than other comparison algorithms and have better learning efficiency of Bayesian Network structure.

Keywords: Dynamic Bayesian Network, BLDOO+CS algorithm, priori network, transfer network, K2 algorithm

1. Introduction

Dynamic Bayesian Networks (DBN) are a type of Bayesian network that incorporates a temporal constraint to better handle time-series data. Currently, DBN has been widely applied in various aspects of production and people's livelihood, such as target threat assessment, Li[1] applied Dynamic Bayesian Networks to evaluate the threat of naval air defense targets; Risk assessment, such as Zhang[2] establishment of a DBN model for comprehensive pipeline gas leakage to evaluate the dynamic risk of pipeline gas leakage; Road surface anomaly detection, such as Li[3] achieving road surface anomaly detection by constructing a high-order Dynamic Bayesian Network classifier; Risk analysis of deep foundation pit construction, risk assessment of ocean going vessels, and analysis and prediction of road traffic safety, such as the deep foundation pit construction risk analysis method

1 School of Mathematics and Data Science, Changji University, Xinjiang Changji, China
2 School of Mathematics and Data Science, Changji University, Xinjiang Changji, China
3 School of Mathematics and Computational Science, Guilin University of Electronic Science and Technology, Guilin, China
*Corresponding author's email: Lfg53880@cjc.edu.cn

based on fuzzy Dynamic Bayesian Network proposed by Shen [4], which solves the construction risk problem caused by the dynamic characteristics of risk factors changing over time during the deep foundation pit construction process; Yingying Wang[5] conducted risk analysis on ocean going vessels by improving the Dynamic Bayesian Network model; Zhao[6] uses the DBN to analyze and predict road traffic safety status; Public opinion crisis warning analysis, such as Tian [7] exploring the level of public opinion crisis in the network of sudden public events by constructing a Dynamic Bayesian Network warning scenario model; The risk assessment of fire and explosion accidents in HRS, such as Lu[8] using the DBN to establish a model for analyzing the dynamic risk of domino accidents in HRS fire and explosion; Financial contagion, such as Fonseca[9] analyzing the phenomenon of financial contagion between economic sectors through Dynamic Bayesian Networks; Dynamic Bayesian Networks have been involved in social phenomena and problems, but the above research mainly focuses on the application of Dynamic Bayesian Networks, with little research on their related algorithms; Some scholars have also conducted research on its algorithm, such as Yang[10] proposing a comprehensive dynamic model that uses mobility data related to residence and work to train and validate the Dynamic Bayesian Network model, with an accuracy rate of 89. 77%, but there was no effective comparison with other related algorithms; Xiao[11] proposed a DBN structure learning algorithm BOA, which first uses various selection mechanisms to obtain a good solution set, optimizes the network graph based on the data information in the good solution set, and then uses the joint distribution of the network graph to generate a new individual set. The new individual solution set replaces the previous generation solution set for iterative optimization of the network structure. Wang[12] proposed an EGA-DBN algorithm, which uses incomplete data as evidence variables to infer missing data, and then calculates the probability of missing data using the concept of mathematical expectation to obtain complete data. Based on the characteristics of the network structure, a DBN structure encoding scheme is designed, and corresponding mutation probability and crossover probability are designed based on genetic algorithm. However, the algorithm is prone to getting stuck in local optima.

Considering that the EGA-DBN algorithm is prone to getting stuck in local optima, this paper proposes a DBN structure optimization algorithm based on BLDOO+CS. The algorithm fully considers the influence of local structure on the final network structure, and proposes a Bayesian Information Criterion (BIC) based local dynamic optimization operator to enable real-time dynamic optimization of the network structure. It is then integrated with the cuckoo algorithm for nest location optimization. The learning of DBN structure is divided into two steps. Firstly, the prior network is learned by adding K2 algorithm to the BLDOO+CS algorithm to obtain the final network structure; Furthermore, for the learning of transfer networks, as the network structure can only point from the current time t to the next time t+1, the BLDOO+CS algorithm is used to learn and modify its structure.

2. Related Basic Theory

2.1 Dynamic Bayesian Networks

Dynamic Bayesian Network is an extension of Bayesian Network in the time domain, which introduces a constraint time attribute to better handle temporal data. DBN can be represented as (B_0, $B_\rightarrow$), the prior network B_0 is a static BN, and the transition network $B_\rightarrow$ refers to the BN of two or more time slices. If only two time slices are considered, the probability of the variable node state occurring at time t is shown in equation (1):

$$P(X^{t+1} \mid X^t) = \prod_{j=1}^{n} P(X_j^t \mid X_{pj}^t) \tag{1}$$

In the above equation, X_i^t represents the value of node j at time t; X_{pj}^t The parent node represented X_i^t ;n represents the number of nodes. The joint probability of any node in the DBN network is shown in equation(2):

$$\begin{aligned} P(X^{0:T}) &= \prod_{j=1}^{n} P_{B_0}(X_j^0 \mid X_{pj}^0) \prod_{t=1}^{T} P_{B_\rightarrow}(X^{t+1} \mid X^t) \\ &= \prod_{j=1}^{n} P_{B_0}(X_j^0 \mid X_{pj}^0) \prod_{t=1}^{n} P(x_i[t] \mid \pi_i[t], \pi_i[t-1]) \end{aligned} \tag{2}$$

Among them, $\prod_{j=1}^{n} P_{B_0}(X_j^0|X_{pj}^0)$ and $\prod_{t=1}^{T} P_{B_\rightarrow}(X^{t+1}|X^t)$ are respectively referred to as the initial network and the transfer network, X_{pj}^0 is the parent node set of X_j^0 in the initial network, $\pi_i[t]$and$\pi_i[t-1]$respectively represent $x_i[t]$belonging to {$X_1[t]$, $X_2[t]$, . . . , $X_n[t]$}in the transfer network, The parent node set $\pi_i[t]$ of $X_n[t]$ and the nodes belonging to {$X_1[t-1]$, $X_2[t-1]$, . . . , $X_n[t-1]$}The parent node set $\pi_i[t-1]$.

2.2 K2 Algorithm

The K2 algorithm [13] is a greedy algorithm that uses node order and maximum number of parent nodes as inputs, and learns network structure from data. The specific ideas about the K2 algorithm can be found in reference [14]. The following definition of node order is shown in equation (3):

$$\mathrm{Pr}ed\left(X_i\right) = X_j = P_a\left(X_i\right) \quad j = 1, 2, \cdots, i-1 \tag{3}$$

Among them, Pred(X_i) represents the node ranked before node X_i, and $P_a(X_i)$ represents the parent node of node X_i.

3. Principles of BLDOO+CS Algorithm Design

Dynamic Bayesian Network prior network learning is based on the K2 algorithm and utilizes the BLDOO+CS algorithm as the core to learn its structure. Determine the maximum number of parent nodes μ by calculating mutual information to construct MWST [15];Using the BLDOO+CS algorithm for iterative optimization to obtain the optimal bird's nest, applying breadth first search strategy [16] to search for node order; Combined with the K2 algorithm to search and obtain the final prior network structure; Based on the transfer network characteristics, the BLDOO+CS algorithm is used to learn and modify its structure.

3.1 Cuckoo Algorithm

Cuckoo Search (CS) is an intelligent algorithm that simulates the special behavior of cuckoo parasitism and brooding to seek optimal solutions. It has the advantages of fewer parameters and is less likely to fall into local optima. A parameter abandonment rate has been developed for the unique habits of cuckoo birds. For the implementation of its algorithm, three ideal states need to be set:

1) Cuckoo can only produce one bird egg at a time and randomly choose other bird nests to hatch its offspring;

2)A randomly selected group of bird nests, and the best nest will be preserved;

3) The number of select able bird nests n is fixed, and the probability that the owner of each nest can detect a sudden addition of a foreign bird egg and discard it is $P_a \in [0, 1]$；

Under the premise of satisfying the above three ideal states, the position update formula for cuckoo nest searching is composed of the current stage nest position plus the Levy distribution function multiplied by the step size scaling factor. The formula is as follows:

$$X_i^{t+1} = X_i^t + \alpha L(s,\lambda) \tag{4}$$

Among them, L(s, λ) represents the Levy distribution function, α is the scaling factor, and X_i^{t+1} represents the position of the nest i at time t+1.

3.2 BLDOO+CS Algorithm

In order to make the cuckoo algorithm more suitable for learning BN structures, several improvements have been proposed: a)Setting up black and white lists and gray lists reduces the search time of the algorithm; b) Inspired by the position update formula for cuckoo nest searching in the cuckoo algorithm, a nest position update formula different from the algorithm was designed to make it more suitable for learning network structures; c) Considering the problem that the advantages and disadvantages of local network structures cannot be generalized, a BIC based local dynamic optimization operator is proposed, and the BLDOO+CS algorithm is iteratively optimized multiple times to obtain the optimal nest position. Considering the problem that the advantages and disadvantages of local network structures cannot be generalized, a BIC based local dynamic optimization operator is proposed, and the BLDOO+CS algorithm is iteratively optimized multiple times to obtain the optimal nest position.

3.2.1 Build Initial Network

Due to the good universality and uniformity of the Maximum Information Coefficient (MIC), for a detailed introduction of MIC, please refer to reference 14. the correlation between nodes is detected by calculating the MIC between nodes, and the MWST is constructed to obtain the network skeleton graph. The initial transition network is constructed based on the correlation between variables. Taking the dynamic ASIA network as an example, as shown in Fig. 1.

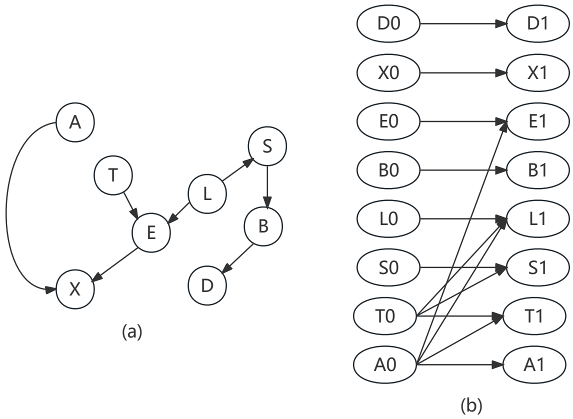

Fig 1: (a)Dynamic ASIA prior network. (b)Dynamic ASIA Transfer Network

3.2.2 Bird's Nest Update

Taking inspiration from the formula for updating the nesting position of cuckoo birds, and considering making it more suitable for BN structure learning, the update formula was improved and a change rule for velocity V was set. The specific formula is as follows:

$$G^{t+1} = G^{t} + V \tag{5}$$

Among them, G^t represents the position of the cuckoo in the nest at time t, G^{t+1} represents the position of the cuckoo in the nest at time t+1, and V represents the cuckoo moving at a certain speed, which is consistent with the definition of the nest position. The velocity V is represented by matrix encoding, where each element is defined as follows.

$$v_{ij} = \begin{cases} 1, & \text{add the edge from node j to node i} \\ 0, & \text{remain unchanged} \\ -1, & \text{delete the edge from node j to node i} \end{cases} \tag{6}$$

When the cuckoo moves from position G^t to another nest position G^{t+1} at a certain speed V, the nest position update is mainly based on the rule setting of the velocity matrix. Taking the survey network structure as an example, the nesting speed V of the cuckoo shown in Fig. 2 indicates that the edge from A→E will be deleted, and the edge from S→R will be added while keeping everything else unchanged.

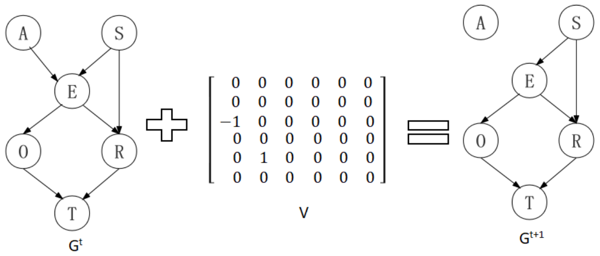

Fig 2 Cuckoo's nest searching method

3.2.3 List division

By calculating the maximum information coefficient between variables as the correlation between their nodes and representing it in the form of a MIC matrix, if setting a single value as the partition threshold unilaterally is too subjective, this article represents the correlation between variables in the form of a percentage. Due to the fact that blacklisting removes independent or unrelated variable pairs, and based on the definition that there is no correlation between variables within the absolute value range of correlation coefficient [0, 0. 09], variable pairs with MIC values less than or equal to 9% are included in the blacklist; Since the whitelist preserves variable pairs with strong correlations, and refers to the definition of strong correlations between variables within the absolute value range of correlation coefficient [0.5, 1], variable pairs with MIC values ranging from the binary to the upper bound are classified as having strong correlations. Therefore, variable pairs with MIC values accounting for 50% or more are included in the whitelist; The remaining variables are recorded in the grey list and subjected to BIC local dynamic selection operator operation.

3.2.4 Based on BIC Local Dynamic Selection Operator

The fitness function of the cuckoo algorithm adopts the Bayesian Information Criterion (BIC) [14] scoring function, which can be decomposed into the sum of scores between each node. Considering that some networks have better local structures but poorer overall structures, such as the current location G^t of the Bird's Nest receiving a better rating than the previous location G^{t-1}, there may be some local structures in G^{t-1} that are better than G^t. Therefore, the speed of cuckoo foraging is determined by evaluating the local scores between nodes. Specifically, the MWST is constructed to obtain the initial network structure, and the variable pairs in the gray list are sorted in descending order. The variable pairs in the gray list are traversed sequentially to change the corresponding speed matrix position from 0 to 1, and the edge with the minimum correlation degree in the parent node set corresponding to the node is deleted. The corresponding speed position is changed from 1 to -1. According to the updated formula, the network Gt is obtained, and the parent node sets of the node in G^{t-1} and G^t are found. The BIC score values of the node and the corresponding parent node set are calculated, and the size of the BIC score values is compared. The node with the larger BIC score value and its parent node set are selected to determine the network structure. G^{t-1} or G^t. Repeat the above operations by traversing through the grey list in sequence.

The definition is based on the BIC Local Dynamic Optimization Operator (BLDOO), BLDOO(G^{t-1}, G^t) is as follows: Input the initial network G^{t-1}, sort the variable pairs in the gray list in descending order, extract the first variable pair (X_i , X_j), set the rules according to the velocity matrix, and change the velocity matrix position v_{ij} from 0 to 1. In G^{t-1}, the velocity position v_{ik} of the variable pair (X_i, X_k) with the lowest correlation degree in the parent node set of node X_i will change from 1 to -1. Based on the updated formula, obtain the network structure G^t. Let the set of parent nodes corresponding $\pi_{X_i}^t$ to node X_i in G^t, the set of parent nodes corresponding $\pi_{X_i}^{t-1}$ to node X_i in G^{t-1}, and the BIC score values $score(X_i,\pi_{X_i}^t)$ of the set of parent nodes $\pi_{X_i}^t$ of node X_i in the bird's nest positions G^{t-1} and G^t be calculated, respectively; The BIC score value $score(X_i,\pi_{X_i}^{t-1})$ of the parent node set of node X_i is denoted as $\pi_{X_i}^{t-1}$. Compare the magnitude of the rating values, if so $score(X_i,\pi_{X_i}^t) \le score(X_i,\pi_{X_i}^{t-1})$, then the set of parent nodes of node X_i remains unchanged; The positions of v_{ij} and v_{ik} in the corresponding velocity matrix remain unchanged, and the position of the bird's nest at the next moment $G^{t+1}=G^{t-1}$; On the contrary, $score(X_i,\pi_{X_i}^t) > score(X_i,\pi_{X_i}^{t-1})$ the position of the bird's nest at the next moment $G^{t+1}=G^t$. Loop through the variables in the gray list one by one until all pairs of variables have been traversed. Finally, check if all the variable pairs in the whitelist are included in the optimized network structure. If they are included, the structure will not change. If not, the edges that are not included will be transformed into the final structure through velocity matrix transformation. The specific operation steps are as follows:

Step1: Sort the grey list in descending order, extract the first variable pair (X_i , X_j), and according to the speed matrix setting rule, the speed matrix position v_{ij} changes from 0 to 1. In the initial network G^{t-1}, the variable pair(X_i , X_k) with the lowest correlation in the parent node set of node X_i corresponds to the speed matrix position v_{ik} changing from 1 to -1. Based on the updated formula, obtain the bird's nest position G^t;

Step2: In G^{t-1}, find the parent node set $\pi_{X_i}^{t-1}$ corresponding to node X_i in G^{t-1};In G^t, find the parent node set $\pi_{X_i}^t$ corresponding to node X_i in G^t;

Step3: Calculate $score(X_i,\pi_{X_i}^t)$ and $score(X_i,\pi_{X_i}^{t-1})$. If $score(X_i,\pi_{X_i}^t) \le score(X_i,\pi_{X_i}^{t-1})$, then the set of parent nodes of node X_i remains unchanged, and the nest position $G^{t+1}=G^{t-1}$ at the next moment; On the contrary, $score(X_i,\pi_{X_i}^t) > score(X_i,\pi_{X_i}^{t-1})$, then take the parent node set of node X_i, and the nest position $G^{t+1}=G^t$ at the next moment;

Step4: Return to Step 1 and iterate through all variable pairs in the grey list until all have been traversed;

Step5: Traverse the variable pairs in the whitelist. If the optimal structure contains all the edges in the whitelist, the structure remains unchanged. Otherwise, the edges that are not included will be transformed through the velocity matrix to obtain the final structure.

3.2.5 BLDOO+CS Algorithm Process Description

Using the cuckoo algorithm as the framework for algorithm improvement, the correlation between variables is first detected, and a network skeleton is constructed based on the correlation and encoded to obtain the initial nest position; Next, update the position of the bird's nest, design a nest update formula, and represent the speed of the cuckoo's nest searching path. The cuckoo moves at a certain speed V while searching for its nest; Finally, the BIC local dynamic optimization operator is proposed to enable real-time dynamic optimization of the network structure, iteratively finding the final nest position. The flowchart is shown in Fig. 3.

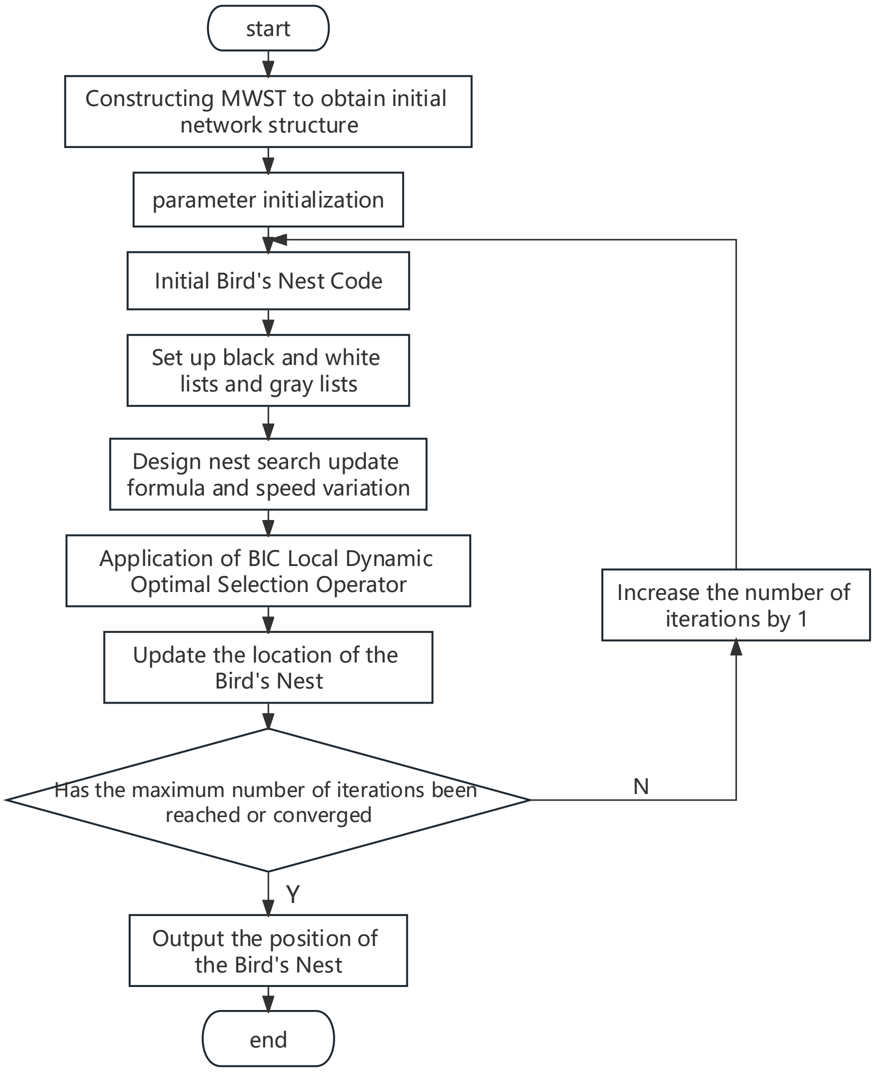

Fig 3 algorithm flow chart

4. Simulation Experiment

4.1 Data Set

The hardware environment for the simulation experiment in this article is Intel Core i7-1087H CPU @ 2. 20GHz, 32GB DDR4 memory, and experimental platform Windows 10 operating system, using Python language for algorithm programming.

To verify the performance of the algorithm proposed in this paper, simulation experiments were conducted using classic dynamic ASIA networks and MILDEW networks as simulation data. Dynamic ASIA Network [17]: Dynamic

ASIA Network simulates data generation by integrating websites https://www.bnlearn.com/bnrepository/.The conditional probabilities between variables in the provided ASIA network were combined with time series to establish dynamic ASIA network simulation data for experimental testing. The dynamic ASIA network structure in this article can be referred to in reference [18].

MILDEW network: The MILDEW[19] dynamic network is a dynamic Bayesian network constructed with winter wheat as the research object and the susceptibility of winter wheat to powdery mildew as the research content. Powdery mildew is a fungal disease that lives on the leaves of living plants, mainly transmitted through wind borne spores. The infection and onset of this disease are highly susceptible to weather conditions, mainly temperature, humidity, and wind. May spread rapidly under favorable weather conditions; Under unfavorable weather conditions for transmission, it may not spread and may disappear over time due to the natural shedding and death of old, infected leaves, while new, uninfected leaves sprout. This is a dynamic Bayesian network that changes over time. The MILDEW network simulates data using conditional probabilities between variables and directly generates data using PYTHON software. Generate simulated data for the ASIA network and MILDEW network, with sample sizes of 500, 1000, 2000, 5000, and 10000, respectively. Compare and analyze the experimental results with the MWST-T-K2 algorithm and MWST-HC algorithm.

4.2 Parameter Setting and Evaluation Indicators

In the simulation experiment, the number of cuckoos was set to n=28, and the algorithm parameters were set as follows: abandonment rate p_a=0.25, and iteration count t=100. Professor Yang [20] found through simulation experiments that the convergence speed is not significantly affected by the values of the parameters p_a and n. After experimental testing, the median value of n in the above range of 15-40 is taken as 28, and p_a is taken as 0.25.

For the evaluation indicators, use Correct Edge (CE) and Hamming Distance (HD), as described in reference [14].

As the learning of DBN structure is divided into two steps, the prior network's correct edge count CE_0 and Hamming distance HD_0 are used, and the transfer network's correct edge count $CE_{\rightarrow}$ and Hamming distance $HD_{\rightarrow}$ are used to judge the quality of the learning structure.

4.3 Analysis of Experimental Results

Record the effective edges (CE_0 and $CE_{\rightarrow}$), lost edges (ME_0 and $ME_{\rightarrow}$), reverse edges (IE_0 and $IE_{\rightarrow}$), and redundant edges (RE_0 and $RE_{\rightarrow}$) of the dynamic ASIA network and MILDEW network in three algorithms. Table 1 records the initial Hamming distance HD_0 and transfer Hamming distance $HD_{\rightarrow}$ learned by the dynamic ASIA network using different algorithms, as well as the correct number of edges CE_0 for the initial network and $CE_{\rightarrow}$ for the transfer network, as shown in Table 1.

Table 1

Comparison of algorithm results of dynamic ASIA networks in different sample sets

algorithm	sample size	HD_0	CE_0	$HD_{\rightarrow}$	$CE_{\rightarrow}$
MWST-HC	500	5	4	7	8
	2000	2	6	5	10
	5000	3	6	5	11
	10000	6	3	3	12
	20000	5	4	3	13
MWST-T-K2	500	3	6	6	11
	2000	1	7	4	12
	5000	1	7	3	13
	10000	1	7	2	14
	20000	1	7	2	15
proposed algorithm	500	4	5	9	12
	2000	3	6	8	13
	5000	2	7	5	14
	10000	1	7	4	15
	20000	0	8	3	15

In Table 1, the learning results of different algorithms in the dynamic ASIA network show that the prior network Hamming distance HD_0 of the algorithm proposed in this paper gradually decreases with the increase of sample size. At a sample size of 10000, the algorithm learned a completely correct prior network; The HD_0 of the MWST-T-K2 algorithm gradually decreases with the increase of sample size, but the inverted edges do not change. Due to errors in the initial undirected graph orientation of the network, the algorithm cannot learn the completely correct network; The MWST-HC algorithm shows an upward and downward fluctuation trend in HD_0 instability as the sample size increases. The correct number of edges for the algorithm proposed in this article gradually increases with the increase of sample size. At sample sizes of 500 and 1000, the correct number of edges learned by the algorithm proposed in this article and the MWST-T-K2 algorithm are 17 and 19, respectively. Therefore, comparing their Hamming distances, the Hamming distances of the algorithm proposed in this article are 13 and 11, respectively, and the Hamming distances of the MWST-T-K2 algorithm are 9 and 5, respectively. Therefore, at sample sizes of 500 and 1000, the learning performance of the MWST-T-K2 algorithm is better than that of the algorithm proposed in this article. The learning performance of the MWST-HC algorithm is lower than that of the algorithm proposed in this paper and the MWST-T-K2 algorithm. Overall, the learning performance of the algorithm proposed in this paper is lower than that of the MWST-T-K2 algorithm when the sample size is small. However, as the sample size increases, the learning performance of the algorithm in this paper is better than other compared algorithms. Moreover, the introduction of the BIC local preference operator makes the edges of the network structure more accurate, filtering out some inaccurate edges in the initial network.

Table 2 records the initial Hamming distance HD_0 and transfer Hamming distance $HD_{\rightarrow}$ learned by the dynamic MILDEW network using different algorithms, as well as the correct number of edges CE_0 for the initial network and $CE_{\rightarrow}$ for the transfer network, as shown in Table 2.

Table 2

Comparison of algorithm results of MILDEW network in different sample sets

algorithm	sample size	HD_0	CE_0	$HD_{\rightarrow}$	$CE_{\rightarrow}$
MWST-HC	500	25	4	13	5
	2000	25	6	11	8
	5000	24	6	9	10
	10000	28	7	8	10
	20000	21	10	10	10
MWST-T-K2	500	34	11	18	11
	2000	28	14	18	11
	5000	22	16	12	14
	10000	22	16	9	14
	20000	21	17	4	16
proposed algorithm	500	21	20	10	13
	2000	18	20	9	14
	5000	18	20	4	16
	10000	15	20	3	17
	20000	9	23	3	17

In Table 2, the learning results of different algorithms in the dynamic MILDEW network show that the HD_0 and $HD_{\rightarrow}$ of the proposed algorithm and MWST-T-K2 algorithm gradually decrease with increasing sample size, but the HD_0 and $HD_{\rightarrow}$ of the proposed algorithm are smaller than those of the MWST-T-K2 algorithm; The MWST-HC algorithm shows an upward and downward fluctuation trend in HD_0 instability with increasing sample size, especially when HD_0 does not decrease but increases at a sample size of 5000, indicating that the algorithm is not stable enough; And the correct number of edges learned by the algorithm proposed in this article is higher than that of the MWST-T-K2 algorithm and MWST-HC algorithm, further verifying the effectiveness of the algorithm proposed in this article.

5. Summary

A new DBN structure learning improvement algorithm is proposed, which divides the black and white lists and gray lists based on the maximum information coefficient ratio between variables on the basis of the cuckoo algorithm. At the same time, a new bird's nest position update formula is designed and the velocity matrix V is designed with rules. Then, a local dynamic optimization operator based on BIC is added to enable real-time dynamic optimization of the network structure, further improving the accuracy of learning the network structure. Finally, the bird's nest position is optimized through the BLDOO+CS algorithm.Experiments have shown that in dynamic ASIA and dynamic MILDEW network data testing, the proposed algorithm outperforms the MWST-HC algorithm and MWST-T-K2 algorithm in terms of HD_0 and $HD_{\rightarrow}$, learning more correct edges than its compared algorithms. It has strong learning ability in terms of structural accuracy and Hamming distance, providing a new research approach for constructing DBN structures.

Acknowledgment

This work was funded by the "Key Laboratory of Financial Big Data" project of Changji University.

REFERENCES

[1] Huiqiang Li, Yinyan Gu, Xinghao Han. Threat Assessment of Ship Air Defense Target Based on Dynamic Cloud Bayesian Network. Shipboard Electronic Countermeasure, 2021, 44(01):38-44. DOI:10. 16426/j. cnki. jcdzdk. 2021. 01. 008.

[2] Jixin Zhang, Dongyang Huang, Qiuju you, et al. Dynamic risk assessment of gas leakage in urban utility based on Dynamic Bayesian Network. Journal of Safety and Environment: 1-11[2023-03-06]. DOI:10. 13637/ j. issn. 1009-6094. 2022. 1554.

[3] Bo Li, Honggang Zhang. Pavement Anomaly Detection Algorithm Based on High-order Dynamic Bayesian Network Embedding. Journal of South China University of Technology: Science Edition, 2020, 48(01):51-59.

[4] Jianhong Shen, Shupeng liu. Application of fuzzy dynamic Bayesian network in risk evolution analysis of deep foundation pit construction. Journal of Safety and Environment: 1-12 [2023-03-06]. DOI:10. 13637/j. issn. 1009-6094. 2022. 2135.

[5] Yingying Wang, Longxia Qian, Mei Hong, et al. Navigation risk assessment for ocean-going ships in the north pacific ocean based on an improved dynamic Bayesian network model. Ocean Engineering, 2025, 315119804-119804.

[6] Pengzhi Zhao, Danyang Geng, Shaoyi She, et al. Road Traffic Safety Status Analysis and Prediction Based on Dynamic Bayesian Network. Journal of Physics: Conference Series, 2024, 2868(1):012028-012028.

[7] Shihai Tian, Yue Yu, Shuyu Deng. Research on the Early Warning of Network Public Opinion Crisis under Multiple Scenarios of Public Emergencies. Information Studies: Theory & Application, 2023, 46(01):132-140. DOI: 10. 16353/j. cnki. 1000 -7490. 2023. 01. 016.

[8] Zihan Lu, Yingbin Cao, Yu Zou, et al. Dynamic risk analysis of fire and explosion domino accidents at hydrogen refueling stations using Dynamic Bayesian Network. International Journal of Hydrogen Energy, 2024, 95546-557.

[9] Fonseca C N, Carvalho F D V J. Analysis of financial contagion among economic sectors through Dynamic Bayesian Networks. Expert Systems With Applications, 2025, 260125448-125448.

[10] Yajie Yang, Soora Rasouli, Feixiong Liao. Integrated modeling of residential and work-related mobilities: A large-scale dynamic Bayesian network approach. Cities, 2025, 156105546-105546.

[11] Qinkun Xiao, Song Gao, Xiaoguang Gao. Constructing DBN structure based on BOA. System Engineering and Electronics, 2007(10):1732-1737.
[12] Fei Wang, Dayou Liu, Yinan Lu, et al. Research on Learning Dynamic Bayesian Networks by Genetic Algorithms. Electronic Journal, 2003(05):698-702.
[13] Behjati S, Beigy H. Improved K2 algorithm for bayesian network structure learning. Engineering Applications of Artificial Intelligence, 2020, 91(05):103617. 1-103617. 12.
[14] Ji Liu, Yuexia Xiong, Lei Li. Bayesian network structure learning algorithm based on MWST-CS-K2. Computer Application Research, 2023, 40(01): 160-164+171. DOI: 10. 19734/j. issn. 1001- 3695. 2022. 06. 0281.
[15] Wenqiang Guo, Lingling Mao, Zixuan Huang, et al. Improved Bayesian Network Structure Learning Based on Evolutionary Algorithm and Its Application. Journal of Henan University of Science and Technology: Science edition, 2022, 43(02): 34-40+6-7.
[16] Yuzhong Wang, Lei Fan, Jianhua Li. Local Community Discovery Algorithm Based on Breadth-first Search. Computer Engineering, 2015, 41(10): 37-41.
[17] Nguyen T. An analysis of East Asian currency area: Bayesian dynamic factor model approach. International Review of Applied Economics, 2010, 24(1):103-117.
[18] Meiting Sun, Bin Liu. DBN Structure Adaptive Learning Algorithm Based on Improved Genetic Algorithms. Journal of Metrology, 2021, 42(01):91-99.
[19] Jensen A L, Jensen F V. MIDAS-an influence diagram for management of mildew in winter wheat/ /Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc. 1996.
[20] Xinshe Yang, Yuxin Zhao, Liqiang Liu. Emerging Metaheuristic Optimization Methods. Beijing: Science Press, 2013.