New constraint-based Bayesian network structure learning method for continuous variables

Chenghao Wei*[a], Yingying Liu[a], Chen Li[a], Chen Song[b], Pukai Wang[a]
[a]School of Computer Science, Hubei University of Technology, Wuhan 430068, Hubei, China;
[b]College of Acupuncture and Moxibustion and Orthopaedics, Hubei University of Chinese Medicine, Wuhan 430065, Hubei, China

ABSTRACT

As an effective tool for knowledge representation and uncertainty reasoning, Bayesian networks (BNs) are widely used in various fields. However, learning the structure of BN is an NP-hard problem. It is impractical to rely solely on the experience and knowledge of domain experts to build BN. Data-driven learning of BN has become a necessity. For the learning of a BN structure with data containing continuous variables, the typical method is to discretize the data or assume that the data follows the Gaussian distribution, and then apply the traditional BN structure learning methods to discover the causal relationship. The discretization inevitably leads to the loss of valuable information of the data. Real-world data sometimes may not follow the Gaussian distribution, which can cause deviation in causality. In this paper, a new constraint-based BN learning method for continuous variables is proposed for BN structure learning. Mutual information and conditional mutual information are derived by a non-parametric kernel density estimation (KDE). The correlation between any two nodes can be determined without assumptions. As new conditional independence tests, they are used in the max-min parents and children (MMPC) algorithm, which is a typical constraint-based method. We compare the proposed method with traditional BN methods using well-known benchmark networks. Synthetic continuous data are generated by linear structural equations. The experimental results show that our method has a good performance. It can be used as an effective BN structure learning method for continuous variables.

Keywords: Bayesian networks, kernel density estimation, constraint-based structure learning, mutual information, conditional mutual information, max-min parents and children

1. INTRODUCTION

Bayesian networks (BNs) play an important role in probabilistic graphical learning. BN can describe qualitative and quantitative dependencies between variables by graphical structural and probabilistic representations, respectively[1]. With a solid theoretical foundation and intuitive representation, BN has a powerful inference capability. It is an effective tool for modeling and reasoning. The method has been widely used in control theory, information retrieval, medical diagnosis, bioinformatics, and computational biology[2].

Unfortunately, BN structure learning is an NP-hard problem. Many BN structure learning algorithms have been proposed, such as sparse candidate (SC)[3], Peter and Clark (PC)[4], three-phase dependency analysis (TPDA)[5,] and max-min parents and children (MMPC)[6]. These algorithms can be effectively implemented for discrete variables, but they are not designed for continuous variables. For continuous variables, a common handling way is to do discretization[7]. This method will not only lose valid information but also may lead to false dependence in the obtained network[8]. Researchers have explored an alternative method, which usually assumes that the samples follow a multivariate Gaussian distribution[9]. L1-regularized Markov blanket (L1MB)[10] and total conditioning (TC)[11] were proposed for BN structure learning under the assumption that the data follow a multivariate Gaussian distribution. However, data in the real world may not follow the multivariate Gaussian distribution. A two-stage algorithm has been proposed for data with non-multivariate Gaussian distributions[12]. This method has a high complexity and limited accuracy during network structure learning.

Kernel density estimation (KDE) is a non-parametric estimation method. As a distribution-free estimation, it has been widely used in various learning tasks[13]. We propose a new MMPC based on the KDE (MMPC-KDE) for continuous variables. The mutual information and conditional mutual information are derived by KDE for conditional independence

*chenghao.wei@hbut.edu.cn

International Conference on Optics, Electronics, and Communication Engineering (OECE 2024), edited by Yang Yue, Proc. of SPIE Vol. 13395, 133951N · © 2024 SPIE · 0277-786X · Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.3048449

testing. It can discover the correlation between different variables and improve the learning effect for continuous variables. As conditional independence testing for continuous variables, they can be used by any constraint-based BN structure learning method.

2. BAYESIAN NETWORK

BN is a probabilistic graphical model that describes the dependence between random variables. The definition of BN can be described by using the directed acyclic graph (DAG). A Bayesian network is a binary set $BN=(\boldsymbol{G},\boldsymbol{\theta})$, where $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ is a DAG, $\boldsymbol{\theta}=\{\theta^1,...,\theta^d\}$ is the conditional probability distribution. Each node in the graph is represented one of the random variables in the set $\boldsymbol{x}=\{x_1,...,x_i,..,x_d\}$. Their correlations are described the edges in the set $\boldsymbol{E}$. The conditional probability distribution for each variable is given by the following equation.

$$\theta^i = P_{x_i} = P(x_i \mid Pa(x_i)) \tag{1}$$

Here $Pa(x_i)$ denotes the set of parent nodes of x_i in $\boldsymbol{G}$. Usually, $BN=(\boldsymbol{G},\boldsymbol{\theta})$ and $BN=(\boldsymbol{G},\{P_{x_i},P_{x_2},...,P_{x_d}\})$ are equivalent. Any BN satisfies the Markov property. By giving the parent node set of any variable, $\forall x_i \in \boldsymbol{x}$, the non-descendant nodes are represented by $ND(x_i)$ and it satisfies the condition $x_i \perp ND(x_i) \mid Pa(x_i)$. According to the Markov property, the joint probability distribution of $BN=(\boldsymbol{G},\boldsymbol{\theta})$ can be simplified as the following equation.

$$P(\boldsymbol{x}) = P(x_1,...,x_d) = \prod_{i=1}^{d} P(x_i \mid Pa(x_i)) \tag{2}$$

3. MUTUAL INFORMATION AND CONDITIONAL MUTUAL INFORMATION USING KDE

A Gaussian kernel is used to smoothly estimate the probability density function, the mutual information and conditional mutual information are obtained by using the estimation. It does not need to assume the distribution form of the data. It is able to handle arbitrary complex distribution shapes. In this section, we offer the principles of KDE first. Then, the mutual information and the conditional mutual information are derived based on the Gaussian kernel function.

3.1 Kernel density estimation

By applying a kernel function to each data point and adding the outcome of the kernel function, KDE can generate a smooth density curve that reflects the distribution characteristics[14]. Assuming samples $\boldsymbol{X}=\{X_1,X_2,...,X_{\mathrm{m}}\}$, $X_k \in R^d$, independently and identically obey an unknown multivariate probability density function *F*. For a given observed vector O, the expression for multivariate KDE is shown below.

$$P(\boldsymbol{o}) = \frac{1}{nh^d}\sum_{k=1}^{m}\prod_{j=1}^{d} K(\frac{o_j - X_{kj}}{h_j}) \tag{3}$$

Here $P(\boldsymbol{o})$ is the probability density function of the *d*-dimensional observer $\boldsymbol{o}$ obtained by KDE. *K*(.)is the kernel function. h is called the smoothing parameter, also known as the window width. *m* is the number of samples. There are many choices of kernel functions for KDE, with the Gaussian kernel function being the most common one. Gaussian kernel function is used for continuous data and it is shown as following equation.

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \tag{4}$$

The expression for the multivariate Gaussian KDE is obtained from equations (3) and (4) as shown in the following equation.

$$P(\boldsymbol{o}) = \frac{1}{n(h\sqrt{2\pi})^d} \sum_{k=1}^{m} \prod_{j=1}^{d} e^{-\frac{(o_j - X_{kj})^2}{2h^2}} \tag{5}$$

3.2 Mutual information and conditional mutual information with Gaussian kernel estimation

Differential entropy and joint differential entropy are measures of uncertainty for continuous random variables and their joint distributions[15]. Given continuous random variables x_1 and x_2, the probability density functions $P(x_1)$ and $P(x_1, x_2)$, differential entropy $H(x_1)$ and joint differential entropy $H(x_1, x_2)$ are defined in the following equation.

$$H(x_1) = -\int P(x_1) \log P(x_1) dx_1 \tag{6}$$

$$H(x_1, x_2) = -\iint P(x_1, x_2) \log P(x_1, x_2) dx_1 dx_2 \tag{7}$$

Mutual information and conditional mutual information can be represented by differential entropy and joint differential entropy, which are the following equation.

$$MI(x_1, x_2) = H(x_1) + H(x_2) - H(x_1, x_2) \tag{8}$$

$$CMI(x_1, x_2 \mid x_3) = H(x_1, x_3) + H(x_2, x_3) - H(x_3) - H(x_1, x_2, x_3) \tag{9}$$

Simplifying the above equation and approximating the mutual information and conditional mutual information using Monte Carlo integrals results for continuous variable.

$$MI(x_1, x_2) = \iint P(x_1, x_2) \log \frac{P(x_1, x_2)}{P(x_1)P(x_2)} dx_1 dx_2 = \frac{1}{n} \sum_{q=1}^{n} \log \frac{P(x_{q1}, x_{q2})}{P(x_{q1})P(x_{q2})} \tag{10}$$

$$CMI(x_1, x_2 \mid x_3) = \iiint P(x_1, x_2, x_3) \log \frac{P(x_1, x_2, x_3)P(x_3)}{P(x_1, x_3)P(x_2, x_3)} dx_1 dx_2 dx_3 = \frac{1}{n} \sum_{q=1}^{n} \log \frac{P(x_{q1}, x_{q2}, x_{q3})P(x_{q3})}{P(x_{q1}, x_{q3})P(x_{q2}, x_{q3})} \tag{11}$$

Here, $P(x_1)$, $P(x_1, x_2)$ and $P(x_1, x_2, x_3)$ can be calculated using equation (5). *n* is the number of the observers. The mutual information and conditional mutual information based on Gaussian KDE are shown in the following equation.

$$MI(x_1, x_2) = \frac{1}{n} \sum_{q=1}^{n} \log \frac{m \sum_{k=1}^{m} e^{-\frac{(x_{q1} - X_{k1})^2 + (x_{q2} - X_{k2})^2}{2h^2}}}{\sum_{k=1}^{m} e^{-\frac{(x_{q1} - X_{k1})^2}{2h^2}} \sum_{k=1}^{m} e^{-\frac{(x_{q2} - X_{k2})^2}{2h^2}}} \tag{12}$$

$$CMI(x_1, x_2 \mid x_3) = \frac{1}{n} \sum_{q=1}^{n} \log \frac{\sum_{k=1}^{m} e^{-\frac{(x_{q1} - X_{k1})^2 + (x_{q2} - X_{k2})^2 + (x_{q3} - X_{k3})^2}{2h^2}} \sum_{k=1}^{m} e^{-\frac{(x_{q3} - X_{k3})^2}{2h^2}}}{\sum_{k=1}^{m} e^{-\frac{(x_{q1} - X_{k1})^2 + (x_{q3} - X_{k3})^2}{2h^2}} \sum_{k=1}^{m} e^{-\frac{(x_{q2} - X_{k2})^2 + (x_{q3} - X_{k3})^2}{2h^2}}} \tag{13}$$

4. CONSTRAINT-BASED BN STRUCTURE LEARNING USING GAUSSIAN KDE

We first discussed the principle of the MMPC algorithm. Then, we use the KDE for independence testing in the MMPC. MMPC is a constraint-based algorithm for learning the structure of BN. The algorithm determines the network structure by discovering independence between variables.

4.1 MMPC algorithm

The MMPC algorithm is a constraint-based local causal discovery algorithm. MMPC views variables as nodes in the graph. The algorithm measures the conditional dependence between two nodes by using the correlation function. A large value indicates that the conditional dependence between the nodes is strong. An edge exists between those two nodes.

When the function value is zero, it indicates that there is no dependency between the two nodes, which means the variables are independent. There will be no edge between the two nodes. An MMPC algorithm includes a growing phase and a shrinking phase. In the growing phase, a Max-Min strategy is used to heuristically introduce nodes into a set of candidate parent-child (*CPC*) for the target node x_t. Specifically, we calculate the minimum correlation value between all other nodes and the target node, and select the node with the highest correlation value to add to the *CPC* of the target node x_t. Given all subsets of *CPC*, if all remaining nodes are independent of the target node, the growing phase will be ended. The shrinking phase mainly removes the variables that should not enter the growing phase. If there is a node x_i in the *CPC* of the target variable x_t, x_i and x_t will be tested for conditional independence with ***S*** as the condition set, and if it is independent, the variable x_i will be removed from the *CPC*.

4.2 MMPC-KDE algorithm

In the MMPC algorithm, the Chi-square testing is usually used to define the correlation function between variables. However, their correlation functions are only applicable to discrete data. Therefore, we will use the mutual information and conditional mutual information based on Gaussian KDE as the correlation function to mention the MMPC-KDE algorithm and the structure of the MMPC-KDE algorithm is shown in Algorithm 1. MMPC can output the undirected graph.

Algorithm 1 MMPC-KDE

```
Input: Dataset $D$; Variable set $x = \{x_1,..., x_d\}$; threshold value $\alpha$
Output: undirected graph $G$
For $x_t \in x$ do
  $CPC(x_t) = \{\}$
  Repeat
    Calculation $AssocF = \text{Max}_{x_i \in x} \text{Min}_{S \subseteq CPC(x_t)} CMI(x_i, x_t | S)$ using equations (12) and (13)
    Calculation $x_F = \arg\text{Max}_{x_i \in x} \text{Min}_{S \subseteq CPC(x_t)} CMI(x_i, x_t | S)$ using equations (12) and (13)
    If $AssocF \neq 0$ then
      $CPC(x_t) = CPC(x_t) \cup x_F$
    Else
      $CPC(x_t) = CPC(x_t)$
    End if
  Until $CPC(x_t)$ has not changed
  For $x_i \in CPC(x_t)$ do
    If $S \subseteq CPC(x_t)$ and $CMI(x_i, x_t | S) < \alpha$ using equations (12) and (13) then
      $CPC(x_t) = CPC(x_t) - \{x_i\}$
    Else
      $CPC(x_t) = CPC(x_t)$
    End if
  End for
End for
Return undirected graph G using the obtained CPC group
```

The skeleton of the network is obtained from the undirected graph. The undirected graph can be turned into a directed graph using orientation rules to correctly discover the dependencies between random variables. The orientation algorithm of Spirtes-Glymour-Scheines[16] is adopted in this paper. As shown in Algorithm 2, the algorithm consists of two steps, the first one is to identify V structure and the second one is to maximize orientation based on constraint propagation. In v-structure, the intermediate node is a common child node of its two parent nodes, and these two parent nodes are independent of each other. When we observe such a structure, we can be certain that the direction of the edge points from the two parent nodes to the middle node. Conversely, if we attempt to change the direction of the edges, it will disrupt the existing v-structure and may introduce unnecessary conditional dependencies.

Algorithm 2 Orientation algorithm

```
Input: undirected graph G
Output: maximize DAG
1. For $x_u - x_z$ and $x_v - x_z$ do
2.    If $x_z \notin S(x_u)\, \delta\delta\, S(x_v)$ then
3.       The directions of the set to $x_u \to x_z$ and $x_v \to x_z$ and gives the V-type structure $V = \{x_u \to x_z \leftarrow x_v\}$
4.    End if
5. End for
6. If G is changed by the following rule, continue, otherwise go to Step 17 then
7.    If $x_u$ and $x_v$ are not adjacent, and $x_u \to x_z - x_v$ is adjacent then
8.       Then the orientation is $x_u \to x_z \to x_v$
9.    End if
10.   If $x_u$ is adjacent to $x_v$ and there is a strictly oriented path from $x_u$ to $x_v$ then
11.      Then set the direction $x_u - x_v$ to $x_u \to x_v$.
12.   End if
13.   If $x_u - x_v$, and simultaneously satisfy $x_u - x_z \to x_v, x_u - x_w \to x_v$ then
14.      Then set the direction $x_u \to x_v$
15.   End if
16. End if
17. Return maximize DAG
```

5. EXPERIMENTAL RESULTS

In the experiment, synthetic data are generated based on linear equations and classical reference BN. We compared the average learning results of traditional discretization MMPC and MMPC-KDE algorithms. Secondly, we also compared the L1MB, Two-phase, TC, and MMPC-KDE in terms of different structural errors.

5.1 Datasets and evaluation metrics

This paper primarily utilizes benchmark networks such as ALARM, CHILD, and ASIA to create datasets for testing and evaluating the algorithm's performance. The three benchmark networks are shown in Table 1. Since the classical benchmark networks are usually used for data with discrete variables, this paper generates the continuous synthetic dataset by linear structural equations and the reference network structure relationship[17].

$$x_i = w_{x_i}^T Pa(x_i) + rand(0,1) \tag{14}$$

Table 1. The benchmark networks.

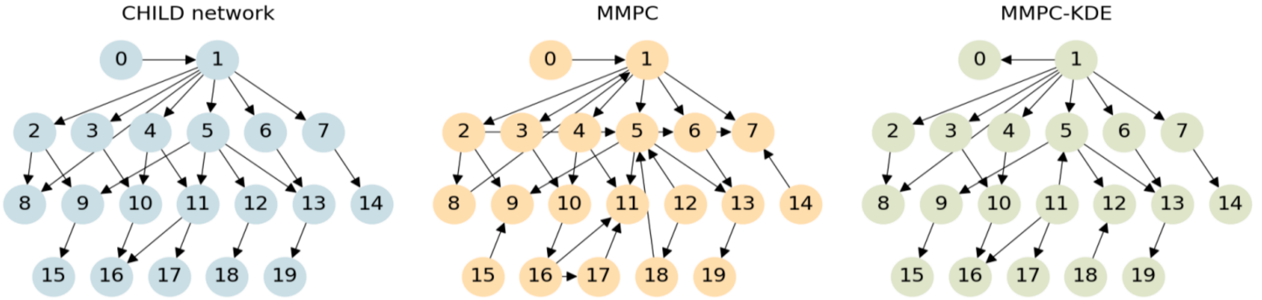
No	Network name	Number of nodes	Number of edges
1	ASIA	8	8
2	CHILD	20	25
3	ALARM	37	46

Here the value of each random variable x_i is a linear combination function of the value of its parent node $Pa(x_i)$ and a random perturbation term. In this equation, $w_{x_i}^T$ is the weight vector of random variable x_i with respect to its parent node, which is generated randomly. The values of each random variable x_i and its parent node $Pa(x_i)$ do not follow a Gaussian distribution. In this paper, we use uniform distribution generation instead of Gaussian distribution. The perturbation terms are also generated by uniform distribution.

We use the metric of structural error to evaluate the effectiveness of the methods. By comparing the obtained network with the reference network, we calculate the structure errors, which include all the errors such as missing edges, extra edges, and reversed edges. The lower the structural errors, the better the network.

5.2 Experimental results

We first compare the MMPC algorithm with the MMPC-KDE algorithm using the benchmark networks. The traditional MMPC algorithm uses the equidistant scattering of the data and the Chi-square testing for BN learning[18]. For each network, we generate datasets with sample sizes of 1000 and 5000, respectively. In the experiments, we set the threshold a to 0.01 and used the classical window width for KDE[19]. Each algorithm is repeated 5 times on the same dataset. The average value of the structural errors is used for the analysis. Figure 1 shows the reference structure of the CHILD network and the optimal learning outcomes of MMPC and MMPC-KDE using a dataset with 5000 samples. Compared with the MMPC algorithm using the Chi-square testing, the missing edges, extra edges, and reversed edges of the BN obtained by the MMPC-KDE algorithm are lower.


CHILD network

MMPC

MMPC-KDE

Figure 1. Comparison of network structures using MMPC and MMPC-KDE on the CHILD network.

Table 2. Experimental results of MMPC and MMPC-KDE algorithm on different datasets.

Dataset	Algorithm	Missing edges	Extra edges	Reversed edges	Structural errors
ASIA-1000	MMPC	3.33	1.00	2.33	6.66
	MMPC-KDE	3.00	0.33	1.33	4.66
ASIA-5000	MMPC	2.67	1.67	1.67	6.01
	MMPC-KDE	2.67	0.00	1.00	3.67
CHILD-1000	MMPC	16.00	3.00	4.00	23.00
	MMPC-KDE	11.33	0.00	3.00	14.33
CHILD-5000	MMPC	9.67	5.00	7.00	21.67
	MMPC-KDE	10.00	0.00	2.33	12.33
ALARM-1000	MMPC	26.33	6.33	5.33	37.99
	MMPC-KDE	24.00	6.00	5.00	35.00
ALARM-5000	MMPC	18.00	8.67	6.67	33.34
	MMPC-KDE	15.33	10.33	3.00	28.66

Table 2 summarizes the experimental results of MMPC and MMPC-KDE algorithms on the different datasets. As can be seen from the table, MMPC-KDE has a smaller structural error than the MMPC algorithm for each network. With the increase in sample size, the structural errors of the algorithm decrease and the network becomes more accurate. To demonstrate the superiority of our algorithm, we compare other BN structure learning algorithms for continuous variables, such as L1MB, Two-phase, and TC algorithms. We recorded the number of structural errors of each algorithm 5 times for three benchmark networks with different samples. Figures 2-4 show the comparison results. In each figure, the horizontal coordinate represents the different data and the vertical coordinate represents the structural errors. For the same reference network, the MMPC-KDE algorithm has a reduction in structural errors as the sample size increases. Compared with the results of the Two-phase, the L1MB and TC algorithms have larger structural errors in all experiments. L1MB and TC do not work for non-Gaussian continuous data. MMPC-KDE is overall better than the two-phase algorithms.

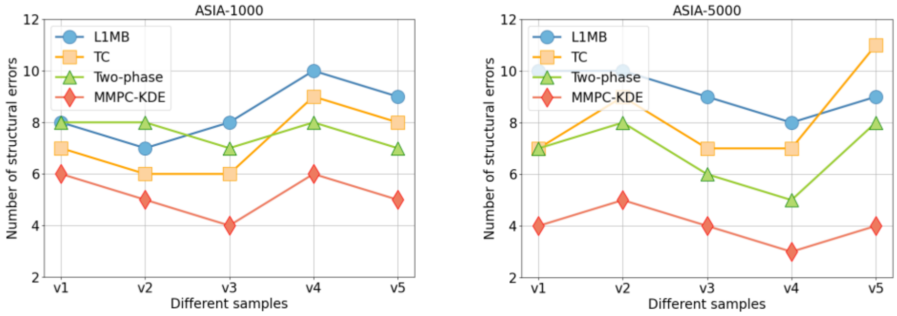


Figure 2. Comparison of structural errors for different algorithms (ASIA).

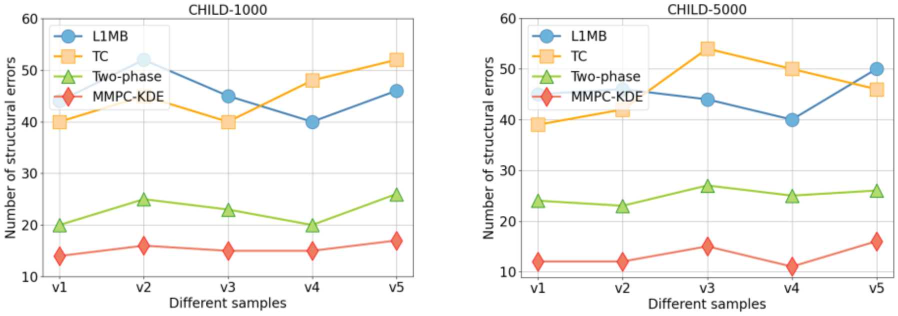


Figure 3. Comparison of structural errors for different algorithms (CHILD).

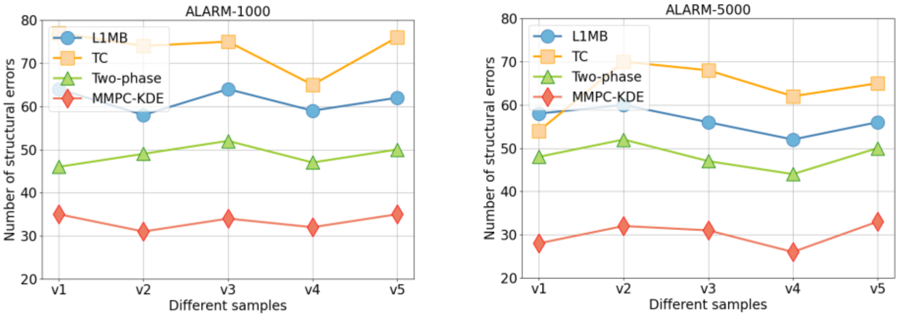


Figure 4. Comparison of structural errors for different algorithms (ALARM).

Non-Gaussian distributed data tend to contain more outliers. The bias correlation coefficient is more sensitive to outliers, and the presence of outliers may lead to inaccurate estimates of the bias correlation coefficient in the Two-phase algorithm. Overall, the MMPC-KDE algorithm demonstrates optimal performance in this experiment, with the smallest structural error.

6. CONCLUSION

This paper proposes a new constraint-based BN structure learning method for continuous variables, which uses the KDE to calculate the mutual information and the conditional mutual information. By using the statistical indicators and MMPC, the accuracy of BN structure learning for continuous variables is improved. This method provides a new conditional independence testing using continuous data. It can be used on any existing constraint-based BN structure learning algorithm. Compared with the original MMPC algorithm, the MMPC-KDE algorithm can achieve higher accuracy. Compared with other BN structure learning algorithms, MMPC-KDE shows higher performance. In future research work, the window width of KDE needs to be optimized to improve the accuracy of BN structure learning.

REFERENCES

[1] Zheng, J., Zeng, L., Yang, Z. and Ge, Z., “Additive dynamic Bayesian networks for enhanced feature learning in soft sensor modeling,” Eng. Appl. Artif. Intell., 126, 108881 (2024).

[2] Ahsan, M., Khan, A., Khan, K. R., Sinha, B. B. and Sharma, A., “Advancements in medical diagnosis and treatment through machine learning: A review,” Expert Syst. J. Knowl. Eng., 41(3), 13499 (2024).

[3] Friedman, N., Nachman, I. and Pe’er, D., “Learning Bayesian network structure from massive datasets: The ‘sparse candidate’ algorithm,” Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, 206-215 (1999).

[4] Spirtes, P. and Glymour, C., “An algorithm for fast recovery of sparse causal graphs,” Social Sci. Comput. Rev., 9(1), 62-72 (1991).

[5] Cheng, J., Grainer, G., Kelly, J., Bell, D. and Lius, W., “Learning Bayesian networks from data: An information-theory based approach,” Artif. Intell., 137(1), 43-90 (2002).

[6] Tsamardinos, I., Aliferis, C. F. and Statnikov, A., “Time and sample efficient discovery of Markov blankets and direct causal relations,” Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 673-678 (2003).

[7] Mabrouk, A., Gonzales, C., Jabet-Chevalier, K. and Chojnaki, E., “Multivariate cluster-based discretization for Bayesian network structure learning,” Proceedings of the 9th International Conference, 155-169 (2015).

[8] Dimitris, M., “Distribution-free learning of Bayesian network structure in continuous domains,” Proceedings of the National Conference on Artificial Intelligence and the Innovative Applications of Artificial Intelligence Conference, pp. 825-830 (2005).

[9] Barros, B., Conde, B., Riveiro, B. and Morales-Nápoles, O., “Gaussian Copula-based Bayesian network approach for characterizing spatial variability in aging steel bridges,” Structural Safety, 106, 102403 (2024).

[10] Schmidt, M., Niculescu-Mizil, A. and Murphy, K., “Learning graphical model structure using L1-regularization paths,” Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1278-1283 (2007).

[11] Pellet, J. P. and Elisseeff, A., “Using Markov blankets for causal structure learning,” J. Mach. Learn. Res., 9(7), 1295-1342 (2008).

[12] Wang, Z. and Chan, R., “An efficient causal discovery algorithm for linear models,” Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1109-1118 (2010).

[13] Chen, J., He, Y., Cheng, Y., Fournier-Viger, P. and Huang, J. Z., “A multiple kernel-based kernel density estimator for multimodal probability density functions,” Eng. Appl. Artif. Intell., 132, 107979 (2024).

[14] Parzen, E., “On estimation of a probability density function and mode,” Annals. Math. Stat., 33(3), 1065-1076 (1962).

[15] Segal, G., Parkinson, D. and Bartlett, S., “Planetary complexity revealed by the joint differential entropy of eigencolors,” Astorn. J., 167, 114 (2024).

[16] Spirtes, P., Glymour, C. and Scheines, R., “From probability to causality,” Philos. Stud., 64(1), 1-36 (1991).

[17] Yang, J., Jiang, L. F., Xie, K., Chen, Q. and Wang, A. G., “Causal structure learning algorithm based on partial rank correlation under additive noise model,” Appl. Artif. Intell., 36(1), 2023390 (2022).

[18] Ma, D. L., Zhou, T., Li, Y., Chen, J. and Huang, Y. P., “Bayesian network analysis of heat transfer deterioration in supercritical water,” Nucl. Eng. Des., 391, 111733 (2022).

[19] Álvarez Chaves, M., Gupta, H. V., Ehret, U. and Guthke, A., “On the accurate estimation of information-theoretic quantities from multi-dimensional sample data,” Entropy, 26(5), 387 (2024).